apcmag.pocketbook

EVERYTHING YOU NEED & NOTHING YOU DON'T

$12.95
From APC Magazine

the Y2K emergency pocketbook

The clear and concise guide to attacking the year 2000 computer problem in business and at home

apcmag.cd THE ESSENTIAL TOOLKIT TO HELP IDENTIFY AND FIX Y2K PROBLEMS

A hype-free guide to finding and eliminating the year 2000 bug

In the book: Mastering the new millennium

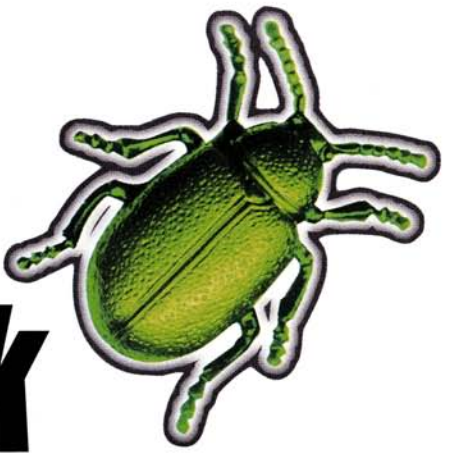
- How to find and eliminate Y2K trouble
- The 'millennium bug' explained
- A guide to every type of Y2K problem
- Last-minute tips for avoiding disaster
- What you can expect in 2000

On the CD: The tools and documents you need

- Testing toolkit: The best Y2K analysis of your hardware and software
- Patches and updates for software and Windows
- Repair app demos: The tools relied on by the professionals
- BIOS tests: Your first step to millennium safety
- Compliance documentation from software and hardware manufacturers

Includes Viasoft Y2K BIOS Test & Fix

See the back of the Pocketbook for details of everything on the CD and inside the book.



the y2k emergency pocketbook

editorial

EDITORIAL

Editor	**Ashton Mills** amills@acp.com.au
Deputy Editor	**Lindsay Hayman** lhayman@acp.com.au
Consulting Editor	**Simon Vandore** svandore@acp.com.au
Sub-Editor	**Vanessa Richardson** vrichard@acp.com.au
Graphic Design	**Niki Creed**
Illustrator	**Ken Rinkel**
Pocketbook CD Editor	**Andrew Broadhead** andrewb@acp.com.au
Contributors	**Angus Kidman** **Michael Smith** **Josh Gliddon**

PUBLISHING

Publishing Director	**Jeremy White** jwhite@acp.com.au
Editorial Director	**Glenn Rees** grees@acp.com.au
Managing Editor	**Richard Rodrigues** rrodrigu@acp.com.au
CP General Manager	**Mike Udabage**
CEO	**John Alexander**
National Advertising Manager	**Mark Harrison** mharriso@acp.com.au
Marketing Manager	**Stephen Dolan** sdolan@acp.com.au
Business Development Manager New Media	**Matt Bateman** mbateman@acp.com.au

For more information about *Australian Personal Computer* or The Y2K Emergency Pocketbook call (02) 9288 9123.

Distributed by Network Distribution Company, 54 Park Street, Sydney, 1028
Telephone: (02) 9282 8777

The Y2K Emergency Pocketbook is published by ACP Computer Publications, a division of ACP Publishing Pty Ltd (ACN 053 273 546)
Printed by Offset Alpine Printing
ISBN 1 876587 06 7

If you're involved in a large business, this book is not for you. By now you hopefully will have taken all the necessary measures to combat Y2K. If you haven't, contingency planning is your best bet at this late stage. If you are a home user or run a small business, however, you still have time to get a grip on Y2K — but you have to act fast. We have called this The Y2K *Emergency* Pocketbook because the deadline is now so close that some people will need to take immediate steps to avoid disaster.

Australia is relatively Y2K-ready, and although it may have been late in alerting people to the problem, we have certainly been bombarded with information during the past year — some of it informative, some of it dismissive and some of it alarmist.

What this book aims to do is give level-headed, useful advice to help you tackle Y2K in a straightforward and practical manner.

While we don't believe Y2K will herald the end of the world as we know it, we certainly acknowledge it will affect everybody in some capacity. It is the extent to which this will happen that remains unclear. However, forewarned is forearmed. If you identify and deal with the areas that are likely to affect you, then you'll have far less chance of running into trouble when 2000 comes around. And whatever does occur will be more easily dealt with than if you hadn't been prepared.

This Pocketbook provides you with some of the background to Y2K — why it's happening and why there's no easy fix. It then goes on to help you identify the key areas with which you should be concerned and provides comprehensive advice on how best to deal with those areas.

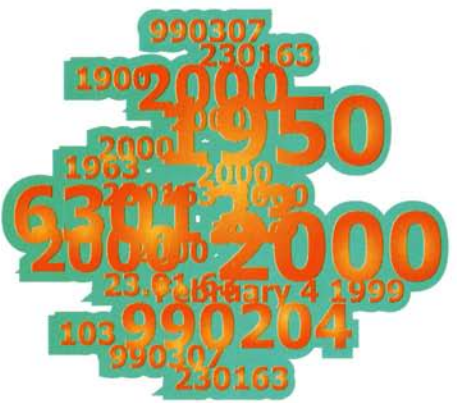
Tackle Y2K before it tackles you, or you may be in for a surprise when you turn on your PC in 2000...

Simon Vandore
Lindsay Hayman

y2k

Introduction

What is the y2k problem?



ALTHOUGH IT HAS BECOME perhaps the most discussed issue in the history of computing, defining the Y2K problem is not as straightforward a task as it may seem. Millions of newspaper articles and casual observers have been happy with explanations along the lines of 'problems caused when computers fail to recognise the year 2000 in dates, assuming instead that it is 1900'; but in itself that definition does nothing to explain why such a problem should have had such a widespread impact.

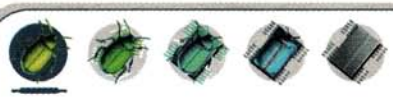
The Y2K problem originates from a very simple decision made by programmers back when computers first began to be used widely in (large) businesses in the late 1950s and early 1960s. Storage was extremely expensive, and processing power had to be used carefully.

To keep information stored to a minimum, numerous new approaches were used to compact data. Names were often stored with only surnames and initials; states were not included, since these could be deduced from postcodes; professions were coded as numbers rather than descriptions.

One of the many decisions made to compact storage at this time concerned the storage of dates. Dates are essential to all kinds of business computing tasks; think of bank statements which must show when transactions were conducted, payroll programs which must operate at a particular time, or interest calculations which are based on the time elapsed from when a loan has commenced.

To keep dates compact, programmers almost invariably chose to store them as a six-digit string, including the day, the month and the year, but not the century. For example, January 23, 1963 might be stored as 230163, or 630123, or 012363, or 23/01/63; the order and the format doesn't matter as long as the program knows which part of the number corresponds to day, month and year.

It's important to note, however, that the Y2K problem itself is not caused by storing dates in this format, but rather due to the way in which that stored information is used to make comparisons and perform calculations. Imagine, for instance, a program which needs to check if a given date is in the future or the past. Say the program wants to check the date February 4, 1999, which it records as 990204, and it knows that today's date is March 7, 1999, recorded as 990307. To determine if the date is in the past, the program can merely subtract 990204 (the query date) from 990307 (the current date). This would give an answer of 103. Since this is a positive number, the date is in the past. If the system performed the same task with the date June



THE Y2K PROBLEM ORIGINATES FROM A VERY SIMPLE DECISION MADE BY PROGRAMMERS

18, 1999 (990618), it would return a negative number (-311), indicating a future date.

So far, so good. But what happens when 2000 rolls around? On January 1, 2000, if the computer performs the same calculation for February 4, 1999, it will subtract 990204 from 000101, and return a negative answer (-990103). The program will deduce from the negative number that the date is in the future, when in fact it is in the past.

This is a somewhat simplified and trivial example used for demonstration purposes. The methods by which dates are stored and compared vary widely, but they will almost always cause problems if there is no century included. (For a more detailed discussion of how calculations can go wrong and the consequences of those errors, see 'The mathematics of Y2K' on page 12).

In some cases, as soon as these programs were created, it was recognised that this compact date format would cause problems in both the short and long term. However, the attitude taken was that the computer systems in use in 1960 (or 1970 or 1980 or 1990) were unlikely to be still in use by the turn of the century.

Obviously, this turned out to be a false assumption. Companies continued to expand their use of computers, but more often than not they built on existing, older systems with new features, rather than starting from scratch. The more data that was stored in these systems, and the more they were used, the less inclined anyone was to go through the very inconvenient process of removing and replacing them. And so the systems stayed in place. It is highly probable that your bank of choice is still running software that is at its heart more than 40 years old, even if your preferred method of access is via much newer technologies such as EFTPOS or the Internet.

Millennium bug or Y2K problem?

WHEN PROBLEMS RELATING to turn-of-the-century dates first became widely discussed, the usual term used was the 'millennium bug'. More recently, the term 'Y2K problem' has gained currency, and the acronym 'Y2K' itself has become a kind of shorthand for the issue. Throughout this book, we will mostly refer to 'Y2K' or the 'Y2K problem'. For other commonly used terms, see the Glossary on page 86.

We've seen it all before, we'll see it all again

THE MILLENNIUM BUG is a widely recognised example of a more general phenomenon: computers which are expecting a certain type of information, and which function incorrectly if they don't receive that information. The technical category into which Y2K falls is known as 'bounded storage', where data is assumed to always be of a certain size or to remain within certain boundaries. This problem in fact occurs with monotonous regularity on computerised systems. Some notable examples include:

- In recent years, the Dow Jones Industrial Average, an aggregate figure which tracks the performance of the US stock market, has frequently moved towards 10,000 points — a historical high. Many computerised systems used by large investment houses have automatic triggers built into them which will give orders to sell, buy or freeze trading when certain conditions are met, such as a large change in the value of the Dow Jones.

  However, many of those systems also originally assumed that the Dow Jones would always fall in a four-digit range (between 0 and 9,999). It was widely feared that if the Dow Jones hit 10,000 points, some computer systems would interpret this as 0, deduce a large drop in the value and sell off huge volumes of stocks, causing fiscal panic and chaos. However, few problems were reported when the Dow finally did top 10,000 in mid-March 1999, which is testimony to the value of early warnings and careful testing.

- Systems which use the Unix operating system generally store dates using a 32-bit digit which represents the number of seconds the date is from midnight on January 1, 1970. This means that there are no Y2K-specific problems with these OSes, but they will run into problems at 03:14:07 on January 19, 2038, which is the date gained by adding the largest possible 32-bit number to the original date in 1970. Companies which shift to 64-bit flavours of Unix can generally avoid this problem, but will still need to conduct rigorous testing. Some Unix firms have incorporated testing for this problem into general Y2K testing (which they still have to do; just because the operating system is broadly compliant, it doesn't mean the applications are).

- Some boundary storage problems occur because certain types of data are explicitly rejected. For instance, programs which use Australian postcodes have often rejected special series postcodes beginning with 1000, 8000 or 9000 (which are used for some large businesses), and most PABX systems had to be adjusted to deal with eight-digit telephone numbers when these were introduced. Because these rules are fairly explicit and often user-selectable, fixing them is often — but not always — straightforward.

The y2k problem: A history

FROM MAINFRAME TO PC

Early computer programs were generally written inhouse by experts and were designed to solve the problems of one specific business. The programs that were created by one company might occasionally be sold to another, but since effective use of IT systems is often viewed as a competitive advantage, companies were reluctant to share this expertise. Once a company was committed to its own in-house solution, it would not shift from it willingly — even if it was riddled with potential Y2K problems.

THE NUMBER OF POSSIBLE PERMUTATIONS OF THE PROBLEM IS INCALCULABLE

While companies such as IBM and Digital began selling some programs for general use in the 1970s, these were still often heavily customised by inhouse IT staff. It was not until the emergence of the PC in the 1980s that mass-market software (designed for use by many different businesses) really took off. Small businesses might have been able to afford a machine for the then cheap price of around $5,000, but they couldn't pay the same again for software. The emergence of cheap spreadsheets and word processors in particular meant that anyone could use a PC, and even customise it to some extent. Suddenly, it seemed, computers were spreading everywhere — and so was the Y2K problem, although few people realised it at the time.

Throughout this period, the programming practices of an earlier age persisted, including the use of compacted dates. Older software continued to use them, while newer programs also often went down this route. Again, this seemed a sensible decision at the time; if your home computer only offers 64,000 characters of storage (for both programs and files) and you want to keep details of 2,000 customers, do you really want to sacrifice 3% of your total available memory every time you enter a date for all of those customers just by including the century?

The real killer, on both PC and other platforms, was the desire to maintain backward compatibility. A key selling point for DOS as it went through its many upgrades was that software written for older versions would continue functioning; a key selling point for every major release of Windows has been the ability to upgrade your OS while keeping your applications. This 'keep your old logic' idea helped PCs become massively popular, but it also meant that fragments of code that dated back 10 or 20 years — to a time when no-one thought storing century data was a necessity — were included in many modern applications. That assumption, that storing century data wasn't necessary, was built into the applications we now use, the operating systems those applications run on and even the processors which our PCs rely on. All are potentially vulnerable.

EVERY SINGLE INSTANCE OF A DATE-RELIANT CALCULATION NEEDS TO BE CHANGED TO ACHIEVE Y2K COMPLIANCE

THE MATHEMATICS OF Y2K

In the introduction, we noted one simple calculation that could go wrong when using compacted dates. The number of possible permutations of the problem is incalculable; dates can be stored in a variety of different formats, and calculations can be performed on those dates in a number of ways, and the results of those calculations used to make other calculations. The end result is little short of chaos, and exceedingly hard to fix even for the original programmer of a piece of code.

While our earlier example used positive or negative numbers to make a comparison, not all applications are encoded for this distinction. A payroll package, for example, might have '92' stored as the year in which a staff member began working for the company. Long-service leave is calculated on the number of years at work, which can be assessed by subtracting the year the staff member started from the current year. Since someone can't have worked at a company for a negative number of years, the original programmer may not have bothered storing the sign (positive or negative) of this number, to save space. In 2000, the number of years at the company will jump from 7 (99-92) to 92 (00-92).

This inaccuracy can then impact other processes. For instance, many payroll systems generate an automatic warning when an excessive amount of leave has been generated. The software might also calculate the cost of leave to the company each week, and start issuing dire warnings about the department where the staff member works. Given that the same problem will afflict anyone who didn't start at the company after 2000, the estimated leave bill may soon appear to be grounds for bankruptcy.

Compliance in a single application is also not enough. For instance, imagine a backup program which is itself Y2K compliant, and which bases its decision on when to perform backups on when a data file was last altered; if the file has changed since the last backup was performed, a new backup is carried out.

If the backup program is running on an older, non-compliant DOS or Windows PC, then the machine is likely to set its system date to January 4, 1980, when it is switched on in 2000 because of DOS's own date assumptions. Other non-compliant programs which alter the file will stamp this as the new date, and when the backup program checks, it will see that the file was altered in 1980, and hence assume it was backed up in a previous session.

These kinds of problems can occur many, many times in any program that makes use of a date in calculations. Every single instance of a date-reliant calculation needs to be changed to achieve Y2K compliance (the term used to indicate a system which can correctly handle dates in 2000 and beyond). This nearly always requires the software to be rewritten — a task

that is generally beyond the individual user (although custom spreadsheets are one obvious exception). In some cases, the only realistic alternative is to upgrade to new software, or even hardware in some instances. Teaching you how to check for these problems — in hardware, operating systems and software — and how to avoid them is the main purpose of this Pocketbook.

DATES TO WATCH OUT FOR

The most commonly cited date for triggering the Y2K problem is, of course, January 1, 2000 (which is, incidentally, a Saturday). As the first date of the new century (for computing purposes), this is when problems will first show up en masse. However, there are a number of other dates in the next two years which will also be affected by the problem.

Some common y2k myths

AS Y2K FEVER HAS SWEPT the globe, a number of popular myths about what's compliant, what isn't and what you can do about it have spread like wildfire. Here are four statements you shouldn't take at face value:

I use a Mac, so I'm OK.
Apple staff have frequently proclaimed that the Mac OS has always been four-digit date aware, so there are no OS-level problems and hence most applications should be OK. Don't believe it. Assuming that your apps will be OK because your OS is OK is a one-way ticket to disaster. And it turns out that one of the Mac's internal routines, StringToDate, doesn't calculate dates correctly (it assumes all dates are in whatever century the Mac's internal clock currently indicates). Check.

I only use word processing software, so I'm OK.
Your word processor may not carry out any date calculations — although most modern WP packages also include quite sophisticated spreadsheet facilities — but it still stamps the date on every file you save. If you use any kind of backup system, this will rely on date stamps to keep track, and will go haywire if your operating system can't cope with dates in 2000. (And if you don't use a backup system, you're a fool.) Check.

I've only just bought a new PC, so I'm OK.
A PC purchased in 1999 shouldn't have too many hardware-level problems, and OS problems will likely be minimal. But what applications are you planning to run? Have you imported spreadsheet data from old systems? Are other machines in your network older? Check.

My business doesn't use a PC, so I'm OK.
This is the biggest myth of all. While you won't need to sort out your inhouse technology, you will need to check out all of your business partners. See 'Dealing with third parties', page 35.

January 1, 1999
This was a key date for travel booking systems, which generally allow data entry up to a year in advance and hence could have invoked a date in 2000 early on. Few problems have been reported to date by travel firms, but then no-one's actually tried to take the flights yet.

July 1, 1999
The beginning of the financial year. Many companies begin preparing basic financial data for the entire year at or around this time (such as entering sales projections and budgets into spreadsheets), and this will be the first time that they will have to deal with dates in 2000. Again, this shouldn't trigger any additional problems to those that would occur in 2000 itself, but it may trigger them earlier. The same will apply later in the year to companies with order systems that look ahead a specific number of days.

September 9, 1999
A common programming practice is to use the sequence of digits 9999 to mark a special function, such as the end of a data sequence or a test date. If software converts September 9, 1999, into the same format (9/9/99), it might have unexpected consequences, ranging from making the record containing the date impossible to delete to making it automatically disappear. This practice was more common in older Cobol systems, but old programming habits die hard.

January 3, 2000
As noted above, January 1 is a Saturday. Many small business computers won't find themselves switched on until January 3 at the earliest, when normal business resumes. All those potential problems that have existed since January 1 will become reality on this date.

February 29, 2000
This is a real date — 2000 is a leap year — but different programmers have made different assumptions about its status, owing to the complex rules of the modern Gregorian calendar. Century years (such as 1900 or 2000) are only leap years if they are divisible by 400 (which 2000 is). The normal leap year rule, which creates a leap year if it is divisible by four, does not apply to century years, even though they are all by definition divisible by four.

Any software routine which calculates the number of days between two dates needs to know about leap years. This is actually a rare example where ignorance may have played in programmers' favour. Programs which assume that division by four is the only leap

Business size
Aware of the Y2K problem
Unaware/don't plan to take action
Intend to take action
Yet to commence Y2K work
Commenced Y2K work
Testing/completed Y2K work

Source: Australian Bureau of Statistics

year rule will correctly deduce the existence of February 29, 2000, although they'll go wrong in 2100. The real concern is programmers who were aware that century years were an exception, but didn't know about the exception to the exception, and who have built in an assumption that 2000 isn't a leap year. A second problem is caused by those especially ignorant programmers who forgot about leap years altogether — although that problem would have shown up in 1996.

October 1, 2000/October 10, 2000

Another, more subtle problem, relates to software which automatically strips out all 'extraneous' digits; for example, the date January 1, 2000, is cut down to 1/1/2000. Even though this includes a four-digit year, some programs can exhibit confusion when they reach October, because it is the first month which must be represented as two digits. Up until October, dates will be have either eight (1/1/2000) or nine (10/1/2000) characters (this assumes the slash is a character). From October 1 (1/10/2000) they will always have nine characters, and from October 10 (10/10/2000) they will have either nine or 10. Not all programs cope equally well with all of these possibilities. Again, if this problem exists it should have become apparent in earlier years, but it's worth checking for it, especially with systems that aren't used on a year-round basis.

January 1, 2100

One cheap-and-dirty method of making software with Y2K problems compliant is to add in a routine which replaces the assumed '19' in front of a date with an assumed '20' under certain circumstances. Such systems might work well throughout next century (although there are far more elegant solutions), but they will go wrong again come the turn of the next century. You might think it's safe to assume that none of those programs will still be in use in 2100, but it was that kind of thinking that caused the Y2K problem in the first place.

WHO'S ACTING, AND WHO ISN'T?

In 1995, hardly anyone had thought about or talked about the Y2K problem, except a

How ready are Australian businesses?

1-4 employees	5-19 employees	20-199 employees	200 or more employees	* All businesses
91	96	99	100	93
46	26	8	1	42
54	74	92	99	58
29	34	28	10	30
12	26	46	62	16
13	15	17	27	13

* Businesses in the agricultural industry are excluded from the business size components, but are included in the 'All Businesses' total.

handful of financial institutions. In 1999, the problem seems unavoidable; it receives coverage not just from the technology press but from mainstream news outlets, and dedicated government agencies promote the importance of Y2K compliance. Despite these massive publicity efforts, however, it seems that we are not responding to Y2K issues as effectively as we should.

Australia is generally considered to have one of the best general levels of Y2K preparation in the world, but this is no cause for complacency. As the table on the previous page shows, business awareness of the Y2K problem is high, but the intention to act is somewhat lower, and large businesses are much more aware of the problem than small ones.

This is bad news for everybody, not just for those companies that haven't chosen to act. As we point out in the next chapter, one of the trickiest areas in Y2K preparation is knowing whether your business partners — the companies you deal with on a day-to-day basis — are also ready for the turn of the century.

Fixing Y2K problems can involve a whole range of tasks. Larger companies need to hire specialist programmers to go through the original code for their applications line by line, looking for potential problems. This is not an especially difficult task — date arithmetic is a fairly simple computing problem — but it is tedious and time-consuming.

AUSTRALIA IS CONSIDERED TO HAVE ONE OF THE BEST GENERAL LEVELS O Y2K PREPARATION IN THE WORLD

In some cases, the date code can be rewritten to include centuries in its calculations, but this also requires all the data used by the program to be updated as well. Other

The Cobol crisis

THE MOST COMMON LANGUAGE that computer programs were written in during the mainframe and mini eras was Cobol (Common Business Oriented Language). While millions of lines of Cobol code remain in active use, the language fell out of favour when PCs became popular (C++ is now predominant in PC application development). As a result, when companies began realising the need to fix older applications, the demand for Cobol programmers far outstripped supply, creating lucrative incomes for many veteran Cobol programmers.

Because Cobol was so widely used in large business systems, it is sometimes assumed that the language is responsible for the preponderance of the Y2K problem. This is not so; it was the way the language was used which created Y2K issues, and the same problems can and do emerge on any platform, with software written in any language.

techniques include making intelligent guesses about which century is implied by a piece of code, although this only works if all the dates given to a program will fall into a span of less than 100 years.

In some cases, companies will decide that the Y2K dependencies built into their existing software are too numerous to fix, and that the only option is to install completely new software systems. Many vendors of software aimed at large businesses have reported major jumps in sales over the past few years, as large firms face up to the need to install new systems or risk Y2K-related problems.

Migrating from a large and complex computer system takes time, however, which is why this is a less realistic option now than it was in 1997.

While small businesses are unlikely to have much custom software, or access to programmers who can check all their code line by line, the fundamental problems faced by them are very much the same, and the choices which they face — repair existing systems or replace them — are identical.

What will happen after 2000?

AS IT STANDS

By the middle of this year, most big businesses should have done everything they could have done to ward off Y2K. Those that haven't will know they're in trouble and will be focusing on contingency planning and workarounds. For individuals and small businesses, however, preparation is still not only possible, it's vital!

The Y2K bug was recognised as a problem well before this year, but it wasn't until 1997 that people really started to wonder 'what does this mean for me, my business and my family?'. By this point, many big businesses had put plans in place to review their exposure to Y2K and had begun to either modify or replace equipment that didn't come up to scratch. They had also started to develop contingency plans and strategies designed to deal with everything from key suppliers and customers suffering from Y2K problems through to the outright failure of major systems like electricity and telecommunications.

Luckily, however, key services like electricity and communications were generally among the first — along with banks and financial institutions — to really get a grip on the Y2K problem and its potential effects.

Testing and improving the plans and modifications designed to cope with Y2K has largely taken place in 1998 and 1999. In Australia most banks, telecommunications companies, utilities and manufacturers are right on track for the full testing and completion of their Y2K efforts by September of this year. After that, it's just a matter of wait and see.

The crucial years for Government preparation and action on Y2K have also been 1998 and 1999. The federal and state governments have the twin tasks of assuring that their own systems are compliant (what would happen, for example, if the tax office's computers suddenly decided that January 1, 2000 was January 1, 1900?), and ensuring that businesses has been active in making their own preparations.

THERE'S A CHANCE THAT SOME COMPANIES MAY EXPERIENCE Y2K PROBLEMS WELL AFTER THE Y2K DEADLIN

The Federal Government has also been instrumental in smoothing the way for businesses to get down to the job of ensuring Y2K compliancy without the spectre of being sued. The Year 2000 Information Disclosure Bill (1999) enables companies to make statements about their level of preparedness for Y2K without worrying that they'll be sued by a customer or a supplier for the contents of the statement. There are a couple of limitations, of course, and it doesn't excuse companies which fail to make Y2K preparations, but it does help everyone know where everyone else is at.

THEN WHAT?

Assuming that we're all still here on January 2, 2000, it should be business as usual for most people, while those who have experienced problems try to implement solutions and make sure they don't impact on the rest of us. That's assuming we haven't all been affected by a common problem such as a utility or telecommunications failure.

But January 2 will not signal the end of Y2K. There's a chance that some companies may experience Y2K problems well after the Y2K deadline, just as some businesses, such as ones that deal with future dates, are beginning to experience them now. Although the chances of Y2K problems recurring throughout the year are lower, there remains a chance that any problems could cause domino effects.

If there is a failure of some sort, it's going to take everyone a little time to get back on their feet, and it'll probably take a while for individuals and businesses to regain confidence. This makes for a good chance that there will be some kind of economic recession to go along with it (see 'What the experts say', page 26).

IS IT FOR REAL?

It is difficult to get your head around something that's as bizarre and yet seemingly obvious, as the Y2K bug. To most people, it seems patently obvious that computers should be able to handle four-digit years — after all, computers are responsible for everything from getting people onto the moon to simulating the monstrous, ghastly environments and creatures in the 'Alien' movies.

But as we've seen, there were genuine reasons for using two-digit years. Back in the day, computers were powerful because they were very conservatively programmed and every single piece of available resources was devoted to doing something vital and useful. And it wasn't out of sheer laziness that the old-time programmers and software developers used two-digit

years. In fact, they considered they were doing their jobs extremely well, and didn't anticipate that they would have done their jobs so well that the machines they designed would still be in use several decades later.

Today, we have the benefit of cheap memory, massive processing power and the luxury of not having to be so economical in the way programs are written. That's why Microsoft Word, for example, is several magnitudes larger than, say, WordStar which was released in 1986, even though both do the same job at largely the same speed.

Massive processing power and cheap memory doesn't make Y2K any less real; nor will today's modern computers simply make the problem go away. The reasons are several fold and quite arcane. First, some computers still need to be economical with their resources. Consider the microcontroller in an elevator. It needs to regulate and convey information about the lift's location, speed and temperature, as well as its next destination, to a central management system. It can't afford to, nor does it need to, have the size, power consumption and resource requirements of a Pentium II processor. As a result, relatively modern buildings may have management systems that still use two-digit dates for reasons of speed and economy. And although these systems are important, it wasn't really a consideration until earlier this decade, when it was realised that date-dependent functions — such as a lift having different movement patterns at night and on weekends — might affect the overall function of the device.

Y2K is also real because old computers don't just go away, and because these small microcontrollers are buried in buildings, factories, cars and aeroplanes in massive numbers across the globe. It's going to be impossible to check all of them, which is why Y2K is going to be more

1960 1965 1970 1975 1980 1985 1990 1995

1960s and 1970s: Mainframe programmers first denote the year in two digits to save memory

1983: First predictions of serious consequences in 2000. Not taken seriously

September 6, 1993: Peter de Jager's wake-up call *Doomsday 2000* published in *Computerworld*. Big business takes notice

1995: Karl Feilder raises Y2K alarm for desktop PCs

serious for small devices than for large ones. Y2K is here, it's real, and it certainly isn't going to go away any time soon.

ARE YOU IN DANGER?

It's pretty hard to get a grip on the threat posed by something as obscure as computer software that doesn't recognise four-digit years. After all, we expect computers to simply keep on doing their job day in, day out whether we're here or not.

Perhaps the best way to think about Y2K is in terms of a massive computer crash. Everyone's familiar with the way that Windows and Macs sometimes just stop working for no apparent reason. And most people will have had this happen just before some critical work was meant to be saved, resulting in a whole lot of wasted effort as the work is re-created.

Y2K is a little like those unexpected PC crashes, except it *could* happen on a massive scale across tens of thousands of computers worldwide. Or, it could be that very little happens at all — a few computers need to be rebooted, a couple of businesses go offline for a day or two and that's it. The reality is that no-one really knows what's going to happen. And that's as accurate as it gets — no-one knows.

Let's assume the worst. Imagine that there's a series of failures in a critical utility that knocks out some telecommunications services. The drop in telecommunications capacity makes it harder for a few financial institutions to keep their electronic transaction systems at full capacity. There might be a run on the banks and massive queues everywhere as panicked citizens try to stock up on bottled water, batteries and tinned food; withdraw cash from the bank; or try to pay bills over the counter.

This scenario isn't really that dangerous, though. Sure it's inconvenient and there are going to be a few frayed tempers, but it is

1997: Height of big business Y2K repair effort

1998: Y2K repair software enters mass market. Many small/medium organisations still unaware

March 1 1999: Most big firms ready. De Jager publishes *Doomsday Avoided*, June 30 1999: Start of 1999-2000 financial year

January 1 2000: Y2K January 3 2000: First business day of new year

2001: Some problems still undergoing repair

doubtful that society is going to collapse. It is more likely that law and order will be maintained, banks will come back online, and utilities will race to fix downed services, should there be any. If you then consider that this is pretty much a worst-case scenario, things don't look too bad. Just make sure that you have a little extra food, shut the front door and watch the fallout on TV.

21BC or the year 23,570. Microwaves and VCRs might be vulnerable, but all they're going to do is get a little confused and think that it's some other date. While this may be annoying if you're trying to record an episode of 'The Simpsons,' it isn't catastrophic.

Cars? Most will be okay because electronic engine management, like the fridge, couldn't care what day it is, so long as the service

ANY MACHINE WITH AN EMBEDDED MICROPROCESSOR THAT USES THE DATE TO HELP DETERMINE ITS FUNCTION IS VULNERABLE

What about the very, very worst-case scenario? What if a combination of Y2K problems causes a domino effect and developing nations lose control of their infrastructure, businesses begin to default on loans, and things gradually trickle up to affect developed nations that have taken Y2K preparations? Well, consider it an outside chance, but anything's possible. In such a situation, only those who are ready for any other big disaster, such as an earthquake or flood, will be able to ease themselves into a situation where widespread hardship is the norm. But in Western nations where a great deal of time and money has been invested in preparations, the chances are law and order will not break down and you will not be in physical danger.

Let's take a look closer to home and consider all those gadgets and devices scattered through our lounge rooms, kitchens and bedrooms.

Will your fridge stop working? Unlikely, as the majority of fridges couldn't care if it were

intervals have been adhered to. In other words, your car should start on January 1, 2000, no questions asked.

In theory, Y2K shouldn't pose any physical threat to anyone, as long as a few simple precautions are taken. Create as much of a buffer as you feel comfortable with, whether that involves extra supplies, raw materials or electricity generators, and remember that New Year's Eve should be a happy time.

COULD IT AFFECT YOU?

The short and easy answer is, yes. Y2K is likely to have some impact on almost everyone, but the extent of that impact is difficult to gauge. There are three spheres where individuals will, in some way, encounter Y2K and its possible effects. Let's consider them in turn.

The private sphere

Home and domestic life is guaranteed to be touched by any Y2K problems in other areas.

This is because most individuals must also function outside the home in some way. Within the home, however, the effects are subtler, and gauging them is an even greater problem than outside the home.

At home, our personal devices — including personal electronics and domestic appliances — are all exposed to Y2K in some way or another. Our phones, for example, are dependent on the service provider doing its job and ensuring its systems are fully compliant and robust enough to withstand the hiccups that might occur in the new year.

Any machine with an embedded microprocessor that uses the date to help determine its function is also vulnerable. That list might include radios, washing machines, microwaves and videos, as well as older PCs, older video games and some hi-fi units. The good news is that most of these units aren't, in fact, so date dependent that they simply won't work. They may get confused, but that's about it.

Y2K will also encroach upon our personal space through its effects on our neighbours, families and friends. However, the personal effects on them won't be any greater than the effects on you, putting everyone in basically the same boat. And remember, our homes are really the least of our worries, because the systems there aren't critical and dependent on big or embedded machines for their instructions.

The public sphere

Here things are a little more interesting because the common services we all depend on — food, water, electricity and transport — are all vulnerable to the effects of Y2K in some way. What's reassuring is that Australian utilities and most big businesses got a handle on their Y2K problems quite early and many analysts believe that there is a low probability here of problems in accessing services after January 1, 2000.

The effect of Y2K on other people will also have a significant impact on the way that public life deals with any potential problems. But if major services are correct in guaranteeing that they will be available without significant interruptions, unrest and disruption are unlikely unless fear itself creates an environment where the general population overreacts.

The business sphere

The impact of Y2K on the business world will be even more interesting, because although most big businesses got their Y2K problems under control long ago, small businesses, in which the majority of Australians are employed, have been notoriously slow in dealing with Y2K.

Consider a retail shop using an old, PC-based point of sale system. It records the type, price and quantity of each sale in a record using the date as one of the key variables for sales comparison. If this system isn't updated, then sales in the new year are going to be incorrect. Worse still, historical sales data may well be rendered unreadable — at least initially — because the system won't be able to determine when the sale actually took place.

There are similar stories scattered throughout small business, and although the impact will be localised, it could be serious for the owners and employees, as well as the neighbourhoods in which these businesses are located, should there be Y2K problems.

The important thing to consider is that Y2K is interdependent on so many other variables that it's really difficult to nail down the effects it may have on any group or individual. It's possible Y2K will have effects that no-one has anticipated, simply because the scope of the problem is so massive. But with a little contingency planning and a dose of luck, most problems can be minimised and localised.

IS IT TOO LATE?

Hopefully we've managed to convince you that while Y2K is a serious problem that you *must* address, it isn't a signal that the end of the world is nigh. You should also be reassured that law and order, water, electricity and financial services are in better shape than some commentators would have you believe, and that there isn't much chance that anarchy will prevail locally on January 1, 2000.

But what if you're a small business owner, or you work for a small business, and no-one's even thought about Y2K yet? Is it too late? The good news is that the answer is, no — it isn't too late for you. If you were involved in a big business, such as a car manufacturer, then things might be a little different. But you're not, so let's see what can be done.

You've taken the first step and bought this book, which is a good start. Follow its advice, turn to our CD and the online resources listed in the back of the book, and you should be in pretty good shape. Ask your colleagues and friends what they're doing about Y2K, and if they laugh, set them straight.

You'll probably also encounter a few doomsayers convinced that the world is about to end or that the Y2K bug is a sign of God's displeasure with humanity. There are also more than a few people who are building bombshelters or bivouacs on remote properties and stocking them with food and water in a bizarre repeat of some Cold War fantasy. Let them believe what they want, and make up your own mind.

Having said that, forewarned is definitely forearmed, and if nothing goes wrong you can always use those extra rations for a big post-Y2K party!

Checkpoint — Background to Y2K

- **Y2K has its origins in the way early computers handled dates**
- **Technically, it's just another 'bounded storage' bug**
- **The bug causes problems in calculations involving dates**
- **Backward compatibility meant Y2K persisted**
- **Private, public and business spheres are all affected**
- **Estimates of its severity vary — but no-one will be immune**

y2k

What the experts say

PLENTY OF Y2K COMMENTATORS are quoted in the media, but you should be wary of trusting the opinions of a single source. Here's a little background information about some of the key figures involved in 2000 speculation.

Peter de Jager

Peter de Jager is generally accepted as having fired the first Y2K warning shot back in 1993 with his now classic *Doomsday 2000* article published in *ComputerWorld* magazine (http://year2000.com/y2kdoomsday.html). At the time, many scoffed and called him alarmist, but today he is seen as the number one authority on Y2K.

His theories are rooted in the idea that Y2K is not just a computer problem, it's a people problem. People are at the root of the tardy response by many companies towards understanding and taking action on Y2K, says de Jager. "Denial of the truth always has a consequence. The time taken to convince people of the severity of the problem was stolen from the time necessary to implement a solution. A shortage of time is something we now have to live with."

He also says that the sooner you try to address a problem, the fewer people are required to fix it. Unfortunately, the time taken by many companies to actually agree that there was a problem, and then to provide resources to fix it, has meant that people who are capable of doing the critical work have been in massively short supply. And because supply and demand are central to the movement of price, the cost of fixing Y2K problems has escalated in a vicious circle — late recognition means more people are required to fix the problem. The number of people capable of fixing the problem is limited, and as a result the cost of fixing the problem escalates. Those who have not yet begun to fix the problem then baulk at the high prices, and the cycle starts over again.

One answer proposed by de Jager was the Office Power User. These are the people who "have usually acquired their skills in direct opposition to office policy. These people were not hired to become computer experts; instead, they became experts simply because they liked using the computer," he said. "No coercion was involved; in fact, it's usually impossible to keep these folks away from computers, even when they should be doing other things around the office."

'DOOMSDAY SCENARIOS' HAVE BEEN AVOIDED. THAT'S GOOD NEWS AND NEEDS TO BE STATED LOUDLY AND STRONGLY

Under this theory, de Jager says that the Power User is generally the person who would understand a key business system best, and

have an almost natural affinity with fixing it. All that would be required is some rudimentary training in programming, and the supervision of someone who actually knows what they're doing, such as a qualified programmer. By implementing a Power User system, businesses would save both time and money, and would be able to circumvent the vicious circle of spiralling costs due to skills shortages.

In March 1999, de Jager published an article entitled *Doomsday Avoided* (http://www.year2000.com/archive /y2ky2kdoomsday.html), in which he said the worst Y2K possibilities had been avoided, though there was still much to be done.

He also warned of the dangers that rampant opportunism poses to those making Y2K preparations. "Here's my assertion," wrote de Jager. "We've avoided global bank failures, global power outages and global communications collapse. These 'doomsday scenarios' have been avoided. That's good news and needs to be stated loudly and strongly. Why? Because there are charlatans and religious extremists masquerading as technical experts and conspiracy theorists posing as computer consultants who are saying everything is going to, with 100% certainty, collapse around our heads. They're wrong and they know it. Trouble is, the media and the average Joe in the street doesn't."

Ed Yardeni

Ed Yardeni (http://www.yardeni.com) is chief economist and global investment strategist for the New York investment house Deutsche Bank Securities. He's also a key figure in Y2K speculation, particularly with regard to the economic consequences of both dealing with the Y2K problem, and the effects caused by it. In a nutshell, his theory is that Y2K could cause a global recession and a 30% reduction in the value of global stock markets. Originally Yardeni said the likelihood of Y2K-induced recession was 70%, and though in recent speeches he has backed off a little, he still believes there is a good chance of global economic deflation caused in part by the Asian economic crisis, and nervousness as the financial world crosses into the new millennium.

YARDENI BELIEVES THAT PROBLEMS IN DEVELOPING NATIONS COULD HAVE A RIPPLE EFFECT ON DEVELOPED ECONOMIES

Yardeni also believes that problems in developing nations, such as Brazil, as well as the countries affected by the Asian Crisis, could have a ripple effect on developed economies like the US, Europe and Australia.

The basis of this ripple effect is the overall interconnectedness of modern economies. "What do you think the effects of not being able to place a phone call in Brazil are?" Yardeni asked in a recent interview. "Brazil's a big exporter of grain; there could be serious disruptions caused by problems just there."

Yardeni has also theorised that many companies are investing too little, too late in their efforts to deal with Y2K, and that this, like the ripple effect from developing nations, could have a significant effect on global economies.

In Australia, the head of the government's year2k Industry Program and the Australian Stock Exchange, Maurice Newman, is also the local head of Deutsche Bank. Yardeni's view of Y2K therefore has some influence here, although Newman is a Y2K commentator in his own right.

Ed Yourdon

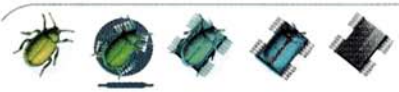
Ed Yourdon (http://www.yourdon.com) is a senior programmer and co-author of the book *Time Bomb 2000*. Central to Yourdon's Y2K theories is the idea of the domino effect, which is that all modern day utilities — and this includes banking, water, electricity and telecommunications — rely on mainframes that are essentially non-compliant. If one of these central systems goes down, then it could, in theory, take down systems that are compliant.

CENTRAL TO YOURDON'S Y2K THEORIES IS THE IDEA OF THE DOMINO EFFECT

Once these compliant systems go down, then business will start to suffer because it can't communicate, can't fulfil orders, and so on. Once business starts to suffer problems, be they big or small, they begin to suffer from a lack of confidence, which reduces economic activity. At the same time, consumers and the public also suffer a crisis in confidence because they see reputable businesses in trouble. This in turn serves to exacerbate the economic slowdown, resulting in recession and, ultimately, a depression that could rival the depression of the 1930s. Grim stuff indeed; Yourdon is regarded as one of the more alarmist mainstream commentators.

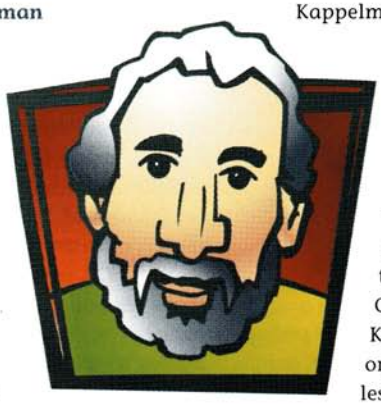
Several sites on the Internet have tried to link Yourdon's book with the views of Dr Gary North (http://garynorth.com), who is the leading prophet of Y2K armageddon. North's predictions of the end of civilisation have been largely discredited — he is known for having predicted nuclear wars and global financial collapses in the past, and has expressed the view that the current world order needs to be changed by some kind of massive disaster. His motivations lie in far-right Christian extremism. Though North is barely mentioned in *Time Bomb 2000*, some of Yourdon's more extreme statements have been compared to those of North. But North is a historian, whereas Yourdon is an expert programmer and his views cannot be as easily dismissed.

(We have included some of Ed Yourdon's essays on the CD under 'Compliance information'.)



THE FACT THAT ANYONE TAKES [SILVER BULLETS] SERIOUSLY IS A SAD REFLECTION OF OUR SCANT UNDERSTANDING OF Y2K PROBLEMS

Professor Leon Kappelman



Professor of Computer Information Systems at the University of North Texas, Leon Kappelman is also co-chairperson of the Society for Information Management Year 2000 Working Group. His primary theory regards 'silver bullets', which in the context of Y2K are methods (such as 'automatic' millennium bug fixing programs) that supposedly rid a system of date-related problems in one fell swoop.

According to Kappelman, however, there are no silver bullets for the Y2K problem. In an email to the Y2K discussion group hosted by de Jager's year2000.com site, Kappelman put it this way: "The fact that anyone takes [silver bullets] seriously is a sad reflection of our scant understanding of Y2K problems. With Y2K we're dealing with a complex set of — interrelated problems [and] THERE WILL NEVER EVER BE A SILVER BULLET" (http://www.garynorth.com/ y2k/ detail_.cfm/2010).

Kappelman has also written extensively on the tardiness of business and government in reacting to Y2K. In a nutshell, his belief is that business and government should focus on systems that are absolutely critical, leaving others for rectification post-2000.

On the other hand, Kappelman believes that organisations are learning lessons from their Y2K efforts which will have long-term benefits. For example, everybody in management now knows much more about the information technology their companies use, and there has been a new type of communication between IT people and management. IT departments have also learned a lot more about outsourcing big projects.

Finally, Kappelman points out that those who are ready for Y2K may find they have some great opportunities in 2000 by taking over the business of those who were not ready. He has predicted "fire sales" by those badly affected, benefiting consumers and business alike.

Feilder's five levels of Y2K

AFTER PETER DE JAGER, Karl Feilder is perhaps the world's second-most prominent Y2K guru. In the early 1990s Feilder started a successful IT company and sold it to Microsoft, then spent time cruising the South African veldt on a Harley Davidson. But it wasn't until he realised that he could be a bad boy *and* stay in IT that he really found his stride.

These days, Feilder travels the globe as CEO of Greenwich Mean Time, the Y2K tools company he founded. He is in demand as a public speaker, and whether or not he's simply trying to drum up sales for GMT's tools, or really does have a philanthropic need to make sure people are aware of how bad Y2K could be, his theories are well worth listening to.

Feilder's theories can be broken into two sections. Part one concerns PCs, and part two concerns the domino effect of Y2K.

As we've already seen, mainframes have hung around in the backrooms of large corporations for years longer than their designers expected them to, and because much of the software was written to use two digits to identify years, Y2K problems have resulted. But Feilder reckons mainframes aren't where the problem lies; instead, he sees PCs as an even greater threat.

On the face of it, that sounds rather strange — after all, mainframes have been around a lot longer than PCs, and mainframes lie at the heart of almost every large corporation. Well, that isn't the case according to Feilder. He is of the opinion that PC software makers have spent a lot of time denying that there are Y2K problems in their products, and that worse still, most of the information in modern corporations isn't stored on mainframes, it's stored on PCs and small PC workgroup servers.

Feilder and Greenwich Mean Time surveyed 1,000 corporations and found that 64% of them were running applications on PCs that were critical to their survival. Only 8% were running such applications only on mainframes, and the other 28% were using midrange systems.

Worse still, Feilder believes only around 4% of the world's PCs have been checked to ensure that they're Y2K compliant — and that doesn't count checking all the old, or legacy data that could have been generated with a non-compliant PC throughout the course of its life.

The Feilder model of compliance starts with the PC, checking that its physical hardware is capable of handling Y2K. Usually that

means examining the computer's BIOS, which is generally hard coded onto a ROM chip, and the battery-powered Real Time Clock (RTC) that provides the system with its baseline time.

The next level after the hardware is your computer's operating system, in most cases Windows, or perhaps Mac OS or a version of Unix. If you're running a Mac you shouldn't have major problems at operating system level, and the same goes for Unix, but older versions of Windows certainly aren't OK — you'll either need to upgrade (from Windows 3.x), or get a patch from the supplier (for Windows 95 and 98).

Then the software needs to be checked. Whereas operating systems and hardware issues are pretty easy to spot, the amount of software that has been programmed over the years makes finding its problems a much bigger task. Different programmers use different techniques to record and deal with dates, so there are no hard and fast rules. One of Greenwich Mean Time's major activities has been to obtain and test thousands of software packages from all over the world in its labs in South Africa and the UK.

FEILDER BELIEVES ONLY AROUND 4% OF THE WORLD'S PCS HAVE BEEN CHECKED

Feilder's fourth level of Y2K is the data created by non-compliant software. Whether it's spreadsheet data, database entries or any other type of recorded information that might involve dates, the prevalence of two-digit years is astounding. Therefore, Feilder estimates that data is perhaps the biggest problem on PCs — it may well be impossible to track down every instance that could affect a company or computer.

Finally, there's the data that is shared across a network, between individual PCs, between customers and suppliers, and between Internet users. Data sourced from someone else and introduced to your system is an unknown quantity when it comes to Y2K — to be confident, checks need to have been performed inside and outside your sphere of influence.

If you've checked your PCs, your data and your network data, what should you do next? According to Feilder, even if your systems are compliant, the domino effect created by the interdependence of systems throughout the world could cause significant problems in the new millennium. The key idea is that the modern world is so interdependent and interrelated that it's almost impossible to figure out how a problem in one part of the system will affect another part of the system. Feilder has predicted that one of the major casualties of the domino effect could be the 2000 Olympics in Sydney.

"I can't see any way that the Olympics will go ahead as planned," Feilder is on record as saying. "It relies on getting people here from all over the place, and in case you hadn't noticed, Australia is a long way away from anywhere."

Only after 2000 will Feilder be judged a hero or charlatan. But no matter which way the cookie crumbles, he will certainly be a lot better off financially.

Embedded systems

COMPUTERS FALL INTO TWO broad categories: general purpose and specific purpose. Personal computers, minicomputers and mainframes can also be described as general purpose computers, because they are able to run a large number of programs. Specific purpose computers, also known as 'embedded systems' or 'embedded chips', are able to run only a single task or a narrowly defined set of tasks.

Embedded systems are used in areas where a general purpose computer would be overkill. They might be found in places such as industrial control equipment governing part of the function of a foundry, or they may be in the anti-lock braking system of a car. Embedded systems can also be found controlling elevators and in the climate control systems within buildings, as well as in medical equipment, automatic doors and consumer equipment like microwaves and video cassette recorders.

Embedded systems fall into three broad categories, depending on their application and the number of additional systems connected to them. These applications also define the complexity of the embedded system, the type of microprocessor used, and the storage that system accesses.

The most basic type of embedded system is the microcontroller. These little processors are generally found within household equipment (like the microwave) and don't have any date storage capability. Microcontrollers generally have their instructions hard coded — or wired — onto

the chip itself. Because they don't handle dates, microcontrollers themselves are unlikely to face any sort of Y2K problems.

Additionally, the systems that rely on microcontrollers are unlikely to have Y2K-related problems, meaning that the chance of them causing a domino effect within related equipment is virtually nil.

The second type of embedded system uses a microprocessor in place of a microcontroller. The defining feature of a microprocessor-controlled embedded system is the use of an internal clock to govern the microprocessor's function.

An example of a microprocessor-type embedded system might be the engine management module in a car that uses the internal clock to correlate service intervals; the controller in the car's anti-lock braking system; or a building management system controlling elevators, air conditioning and alarms in a high-rise apartment block.

Because microprocessor-controlled embedded systems use an internal clock, they are more likely to fail due to Y2K-related problems. The upside is that if these type of embedded systems are isolated from other systems, they're likely to go into standby mode, shutting down the equipment they control.

The UK's Institution of Electrical Engineers (IEE) estimates only one in 100 microprocessor-controlled embedded systems failures will have a significant business impact. This has proved accurate in most Y2K testing of embedded chips to date. The IEE also estimates that there will be a larger number of actual system failures, but these failures won't be critical because they operate independently — that is, there aren't associated systems relying on the output of the embedded system that has failed.

The final type of embedded system is what is known as a large-scale system. Typically, these involve a number of discrete but interdependent systems that are overseen by a general purpose machine. The IEE ranks these interdependent systems as most vulnerable to failure, and estimates that because they use complex application software, the failure is likely to be software related. The failure of a single chip in one such group can create havoc, because a standard instruction in such chips is along the lines of 'if there is any kind of problem, stop'.

Interdependent systems are also unique in that they can use a hard drive for storage. Other embedded systems are entirely contained within instructions that are hard coded into the processor itself, or hard coded into an associated memory chip.

IDENTIFYING PROBLEMS

Because they're ubiquitous, the failure of embedded systems carries a significant risk to businesses at all ends of the spectrum. They're unlikely to cause such problems in your home, though occupants of modern high-rise buildings should ensure their building is tested.

In an organisation, once an inventory

and basic testing has been completed, the supplier of the system should be contacted in order to a) verify the test results and b) obtain possible solutions and/or workarounds for any likely Y2K problems. The results of testing and information obtained from the system supplier should then be plotted within a compliance database.

It's worth building a complete database of every system within the organisation containing information about the system, its test results, information from the supplier and its compliance status. The database should also contain data relating to the risk that the failure of the system could pose to the organisation.

There are four options for non-compliant systems: they could either be repaired, or thrown out and replaced; a workaround could be put in place; or the system could be phased out altogether, if it no longer had a significant business function.

Repairing the system can pose the greatest challenge because of the nature of embedded systems, and involves removing and replacing the physical instructions that are hard coded onto the processor or its associated memory. It is only really an option where it is easy to do so, or where non repair, such as in the case of a supplier of systems incorporating embedded elements, could result in the owner or manufacturer being held legally liable if the system fails in the line of duty.

Embedded systems are always part of a bigger picture, so for the purposes of Y2K it's important to understand the context in which an embedded system carries out its tasks — often, assumptions are made by both the system operator and the system's software.

The IEE illustrates assumptive behaviour as follows: a chip assumes that because a sprinkler system is operating, a fire will be put out. However, there may be no water in the pipes, rendering the firefighting capabilities of the system useless.

Checkpoint — Embedded systems risks

- Embedded chips are used where a computer would be overkill
- They are found in many types of electrical equipment
- Microcontrollers, the most basic, pose a low risk
- Microprocessors are more vulnerable to Y2K
- Large-scale embedded systems are the main danger area
- Interdependence of embedded chips can cause them all to stop

Dealing with third parties

ONE OF THE MOST important facets of the Y2K problem is also one that has only received attention in relatively recent times: the fact that other people's Y2K problems can impact your business. No matter how much effort you put into checking your own internal IT systems, if one of your suppliers or customers has a major problem with Y2K, that will have a flow-on effect.

This problem has become especially pronounced in recent years as the interchange of information via electronic systems has become common. While exchanging data over the Internet is a recent development, banks have been receiving payroll instructions from employers on disk for much longer. If either end of the system isn't working, then the entire process collapses.

It's a mistake to assume that you're safe simply because you don't exchange information electronically. Even if your only contact with the freight company that delivers bread to your shop every morning is signing off an invoice, that invoice has still probably been produced on a computer. And even if the invoicing application itself uses dates intelligently, that won't help if the bakery's internal network is down because

So long

IT ISN'T REALISTIC TO ASSUME that you'll be able to convince every business that you deal with to do something about potential Y2K problems if they appear to be lax. If, however, one of your major suppliers consistently refuses to provide information about their activities in this area, you have little choice but to look elsewhere. This is easier said than done in some cases — most rural consumers can't change telephone providers easily, for instance, and if you have a need for unique parts or services then switching may be more difficult — but you need to consider carefully the consequences of continuing to work with a company that may go 'off the air' for an extended period of time when 2000 kicks in.

it wasn't able to cope with the 21st century. Partial compliance is no solution.

It's essential that you check with all your major business contacts to ensure that they are also ready to deal with the Y2K problem. You need to pay particular attention to the following three categories:

Suppliers

Suppliers range from the firms that supply you with raw materials to service providers in areas such as transport and communications. You're not likely to forget to check your single largest component supplier — but have you checked your accountant? Anyone who makes a major contribution to helping your business run can also make a major contribution to stopping it dead in its tracks.

Customers

This may seem like a less important area — until you remember that from the customer's point of view, you're the supplier. And if you have a number of large customers whose business you rely on, there'll be problems for your cashflow if they suddenly suspend activities for a month due to unforeseen problems. You may feel awkward about asking your major customers if they are Y2K compliant; explaining your own activities in this area might be one good way into the conversation.

Export markets

If you send products overseas, then the range of issues you need to check is much broader. Are the customs services in countries you export to Y2K compliant? What about air traffic control systems, or freight tracking? Do you need to export extra products early to cover yourself in case a problem emerges? Can you change your trading practices to lessen the risk? Levels of preparation vary widely in different world markets, as the accompanying map indicates.

In all three areas, you will have to make a choice if it emerges that a third party is not going to be ready in time. In extreme cases,

you may need to switch suppliers. A less drastic alternative is to make a contingency plan to deal with areas where a failure might occur (see page 65).

You also need to be ready to disclose information about your own level of Y2K preparedness to others. When explaining the activities you've undertaken, don't be tempted to lie or exaggerate. If you've tested carefully and thought through all the possible consequences, you have nothing to fear from honesty.

Large enterprises have again been forced to take a stand on these issues earlier. The Australian Stock Exchange, for instance, requires regular statements to be issued by all listed companies on how they are progressing in dealing with the Y2K problem. Failure to issue such a statement can result in removal from the exchange. The ASX compliance regime has also produced some sobering statistics; one analysis of figures submitted last year found that the total cost of Y2K compliance for Australia's top 150 companies would top $4 billion.

Surprisingly, though, not all companies have published these statements in any format other than a statement to the ASX. A survey by the Australian Securities and Investment Commission (ASIC) found that almost 60% of publicly listed companies had not included Y2K compliance information in their annual reports — a poor example to set to others.

To encourage companies to make free and full disclosures of their Y2K compliance efforts, the Federal Government earlier this year passed the Year 2000 Information Disclosure Bill (often referred to as 'good Samaritan' legislation). This has eliminated the lingering concern for many businesses that revealing they had not yet completed (or begun) Y2K preparations would leave them open to lawsuits.

While companies can still be sued for defective products or services resulting from a Y2K glitch (a chastening thought in itself), statements regarding Y2K preparations can't be used as evidence. The bill runs through until the middle of 2001, a sensible recognition that Y2K issues will continue to affect us for some time in the next millennium. To encourage Y2K fixes, taxation law has also been changed to allow alterations to make software compliant allowable as an immediate deduction.

Checkpoint — Third party plans

- **Make sure your main business contacts are also Y2K ready**
- **If they are not, decide on a contingency plan**
- **If they will not provide information, change supplier**
- **Be ready to disclose your own level of preparedness**

Can I get away with doing nothing?

A VERY BASIC Y2K rule of thumb goes as follows: If you use a computer, or a computerised device, anywhere in your business, you need to check that it's compliant. Even if your company doesn't use a computer, many of the essential services which you rely on — banking, telephone, electricity, even the post — do. If they go down, then you will definitely go down with them.

IT WOULD BE AN EXAGGERATION TO SUGGEST THAT EVERY SINGLE PC IN THE WORLD WILL CEASE TO BECOME FUNCTIONAL

However, it would be an exaggeration to suggest that every single PC in the world will cease to become functional on January 1, 2000. If you have a functional Commodore 64 which you regularly use just for word processing, and you're happy with what it offers, then there's no need to upgrade your machine. Similarly, if you use an XT or AT system, conduct your backups manually and are working productively, then your need to upgrade may be minimal if you don't use date-sensitive applications. Any PC which is used largely for gaming is also likely to suffer few problems, although this may not apply if you want to compete online with other players.

It's also important to recognise that there are classes of computer-like devices for which the problem is in some ways irrelevant. Systems which use 'thin clients' — low-spec PCs which download their OS and software from a central network, rather than their local drives — won't face the same difficulties in upgrading, for instance. As long as the server itself has been made compliant, the individual systems should continue functioning.

Essentially, then, there are two key principles which you need to follow. The first is to recognise that while you may not choose to take any action to make your current PC compliant, you do need to take the time to assess whether there may be a problem. Once you've checked, you can make the decision on whether to upgrade hardware or software or stick with what you have, but it will be an *informed* decision.

The second is that if you do choose to continue using a machine which is not compliant, you should take steps to ensure that you don't accidentally (or deliberately) distribute bad data to other users. In particular, such machines should be kept well away from networks of all types, and you should warn other users with whom you exchange data (even if it's just on a floppy) that there may be problems with the data's current format.

Attacking the problem

Attacking the problem

WHERE TO START

SO, YOU NOW KNOW HOW Y2K originated, where it's likely to appear and the ways in which it might affect you. But what can you do about it?

Most people's natural reaction is to rush headlong into fixing everything in sight, as quickly as possible. Hold it! The professionals now recommend that you begin with a bit of navel-gazing to understand the scope of your problem, to ensure that your efforts and remaining time are allocated to the most important areas.

If you run a business of any size, ask yourself and your colleagues the following questions: Why does the company exist? How does it work? How is technology involved? What is the potential impact of technology failure inside and outside your organisation?

If you're an individual fixing your own computer(s), you should ask yourself similar questions: Why do you have a computer? In what ways do you rely on it? What if you could not rely on it?

Some of the answers might seem obvious, but these 'what if' situations should make it clearer where to start. For example, if a business depends on computers to instruct and monitor machinery, the most important area to fix might be the instructing systems.

Monitoring is very important, but if need be, the machinery could probably run without it; whereas it can't do a thing in the absence of instructions. For example, the electricity grid in New South Wales is not IT-intensive and it is expected that the risk of power interruptions is low, but it is monitored by systems which have required extensive Y2K checking.

If you use your computer to play games, to work from home and to make Web pages about a hobby, you should focus first on the work side of things. If your employer is sensible, some help may even be available to ensure your PC is compliant. If you are self-employed, you need to treat the situation with the same seriousness as any other business. On the other hand, people across the world may be relying on the information you share about your hobby, so it had better be Y2K ready! And yes, games can also be affected, perhaps by operating system inadequacies (though this is unlikely, as the most popular games are generally relatively new and require recent equipment).

YOUR COURSE OF ACTION DEPENDS ON THE SIZE OF YOUR PROJECT AND THE TIMEFRAME

Your course of action depends on the size of your project and the timeframe. If it's December 1999 and it looks like your accounting system will fail in 2000, don't rush around tearing out your hair or scrambling to find a replacement product; move straight to contingency planning based on what you know about how the accounting system affects everything else. Awareness is a huge advantage, and though there will no doubt be adverse effects, these will be reduced by your preparedness. Imagine if such a thing happened out of the blue!

The term used in the corporate world for this kind of introspection and direction is 'business continuity'. Without the means of doing business and earning money, an organisation is in deep trouble. Business continuity involves ensuring that a firm's vital operations can keep ploughing along despite adversity, by singling out the core processes and their weaknesses well in advance. This may involve such things as stockpiling resources, arranging fall-back premises (such as employees working from home), or placing an emphasis on decentralised operations, such as a virtual shopfront on the Web. There are even a few continuity consultants around, who specialise in developing such contingency plans (see page 65) and helping companies recover from disasters — business continuity is about more than just Y2K.

As if in warning, there were three occasions in our region in 1998 when business continuity became an immediate concern: the Auckland power blackouts, the Sydney water scare and the Victorian gas crisis. Each affected the way people went about their daily lives, and posed threats to the way companies did their daily business.

The Auckland example was perhaps the most akin to Y2K's potential effects, as it directly affected computer usage. The entire central business district (CBD) of a major city was left without electricity for the better part of a month — a situation which required all businesses housed there to reinvent themselves. Some managed to run on the limited power of fuelled generators, a few had their employees work from home, and many found temporary office space outside the CBD. Shops and services were the hardest hit; tied to a location and deprived of customers, some simply had to sit it out.

Everyone suffered in one way or another (except perhaps generator manufacturers), and compensation was required. But in some cases those who successfully adapted were a little reluctant to go back to the old way of doing things. The definitions of organisations had changed, and their options had broadened.

Once you've clarified the purpose of your business and how technology affects that, you'll be ready to look at the Y2K situation on the ground.

Checkpoint — Plan of attack

- **Don't rush into a fixing frenzy**
- **Begin by taking stock of your situation**
- **The size of your problem and the time you have left will determine what steps you take**
- **Business continuity should be your main focus if time runs out**

Taking an inventory

THIS PROCESS ISN'T TOO HARD if you're an individual PC owner. You can probably rattle off in seconds the major components of your computer and the software installed on it. You know which apps are crucial and the data on your computer is neatly arranged in your personal document folders. Maybe you have a second computer and some network hardware — but you know where it is and what it does.

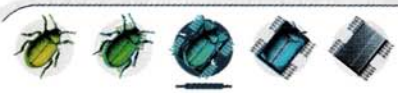
Spare a thought, though, for the IT managers of large corporations; as much as they try, they cannot be familiar with every little piece of hardware on or under every desk in the organisation, let alone what everyone's doing with it and where every little bit of non-compliant data lives. It's a big ask. On the other hand, it's part of an IT manager's job, so the groundwork should already be there and the status of most items should already be tracked by software on the network.

As many Y2K problems are a result of the interdependence of several items, it's very important to know exactly what you've got; often there's a long-forgotten 386 sitting under someone's desk, performing minor but time-critical server duties and waiting to die when its clock says 00.

The inventory and assessment stage of Y2K work is the easiest to complete — implementation and testing can take much longer — but companies of any size should approach making an inventory at this late stage with great caution. Deciding whether or not to make an inventory and begin manual repair is highly dependent on the time you have available. We've heard Y2K consultants say they wouldn't even consider taking an inventory at this stage, and would instead use the limited time frame to plan for emergencies and quickly patch up problems that are immediately visible.

If taking an inventory is still worthwhile in your situation, remember that the purpose of the exercise is to categorise everything by the potential type of problem it could create. It may be worth giving each item a score out of 10 to indicate severity.

It will be necessary to record manufacturer and version numbers for all software, hardware and firmware (the BIOS version, for example, is shown briefly when a PC boots). The most obvious bugbears are older applications and hardware, which often don't have a 2000-compatible version, unlike their modern counterparts which may have patches or additions which are available from the manufacturer.



Don't ignore newer items — even Windows 98 needs a couple of Y2K patches, and you should check the compliance of other off-the-shelf software with the product's manufacturer.

A more difficult area is proprietary software — anything written specifically by or for your own operation. Often the programmer would be the best person to fix such code, but they may not be available, or the application may be a mishmash of input from various people.

If you're using a spreadsheet to record this data, you might like to refer to a sample Y2K inventory form on the Web. Examples include:

- **http://www.y2krun.com/ch1chk.htm**
- **http://www.amt-mep.org/y2k-forms.htm**
- **http://www1.cc.emory.edu/ITD/YEAR2000/y2k_inv_templ.html**

CONSIDER AUTOMATION

If you are taking an inventory of a large amount of equipment, then you should really think about automation rather than simply using a spreadsheet. There is plenty of inventory software available on the market. Viasoft's OnMark WebCenter has an innovative Web-style interface, and a lower-cost option is the shareware Millennium Manager 2000 (http://www.year2000software.com).

Not all information can be gathered automatically; sometimes the best idea is to consult the people who use a particular piece of equipment.

In a dynamic environment you will probably be adding and removing programs and hardware all the time; be careful not to introduce new Y2K problems this way. The same goes for sharing data or technology with suppliers or customers.

Checkpoint — Take stock

- **Before you begin, decide whether you have the time and resources to complete the task**
- **If not, switch to contingency planning**
- **Record the potential severity of each component**
- **Record manufacturer and version numbers for all software, hardware and firmware**
- **Older apps and hardware will need most appraisal**
- **Don't ignore recent purchases**
- **If possible, contact the programmers of proprietary software**

Repair process

THE REPAIR PROCESS

Y2K repair, or 'remediation' as the professionals call it, involves tracking down a lot of small problems to prevent a bigger one. As we have seen, a simple date-related problem can have serious consequences when the data it produces is used in a computing process, whether directly or further down the line from where it originated.

Many of the tasks involved in remediation are very simple and repetitive; the complexity lies in managing such an assortment of simple tasks. Effective remediation requires a capacity for lateral thinking, for which Y2K experts are paid a great deal of money; but handling things by yourself at home or in a smallish business should not present too much of a problem, if you pick the right tools for your particular situation.

Every analyst, consultancy and commentator seems to have a different estimate of what the total cost of Y2K work will be for PC owners worldwide (it's usually in the tens of billions of dollars). It's certainly an undesirable expense, but history shows that the mess created by one computer error can be even more expensive, so the idea is to spend what you must now to minimise risks.

Hopefully by now you will have examined how you use technology and what would happen if it failed, so you will be in a position to decide where to start, taking into account the time that remains before 2000. Remember that remediation is not necessarily the first or most important step.

THE BIOS AND REAL-TIME CLOCK

All hardware has some kind of BIOS or 'firmware' stored in read-only memory that gives it basic instructions on what to do. Part of the BIOS is generally the system clock, which lets the operating system and software know what time it is. But the system clock itself gets its time from a battery-operated 'real-time clock' (RTC) in the CMOS. The RTC keeps time whether the hardware is switched on or off.

Modern real-time clocks have a field representing the century, but what's most important here is that your BIOS is Y2K compliant — that it works with years in four digits, not two, and that it therefore knows what to do with the RTC information when 2000 arrives. Even if your RTC can only handle 99 years (which is quite common), a compliant BIOS should be able to compensate for this in the system clock.

Note that there is some disagreement across the computer industry as to the importance of RTC compliance, because while it is actually possible for a piece of software to bypass the system clock and get its date from the real-time clock, this is very unusual and completely against industry standards. So there's a risk in having a non-compliant RTC, but estimates of its severity vary. All mass-market software should get its time information from the BIOS, but if you run a business involving proprietary manufacturing or financial systems that need very precise timing information, it's possible that the RTC will need to be replaced.

There is also some concern about the Crouch-Echlin effect, which occurs when extra BIOS routines associated with post-2000 dates cause further timing problems when PCs (especially older machines) are switched on. The solution to Crouch-Echlin is a modern, buffered RTC. However, Intel claims to have disproved Crouch-Echlin (see http://www.intel.com.support/year2000/c-e-wp.htm).

HANDLING THINGS BY YOURSELF AT HOME SHOULD NOT PRESENT TOO MUCH OF A PROBLEM

The safest way to test BIOS compliancy is to use an automatic date rollover program from a reputable supplier. There are dozens available, including many on the Internet (you might want to try a few different ones just be to be sure), and on the Pocketbook CD, or at ridiculous prices in the shops. Generally they create a controlled environment in which your system clock goes from 1999 to 2000. Some will test for other issues like the

THE SAFEST WAY TO TEST BIOS COMPLIANCY IS TO USE AN AUTOMATIC DATE ROLLOVER PROGRAM

leap year in 2000 and the 9/9/99 problem. Some PCs' compliance behaviour will vary depending on whether the machine is switched on or switched off due to real-time clock issues. A few BIOS testing programs will also test this; they require you to power down your machine for a minute or so. A few more advanced testing tools for networks allow administrators to test PCs remotely, and to find firmware details across the network.

If you were to do such tests manually and boot up your PC with the wrong date, any time-critical processes you ran might be affected — especially if you temporarily forget to change the date back when you reboot! Automated checking is faster and will generally give you a plain-English report on the situation, rather than leaving you to figure out the correlation between certain situations and reactions.

If you aren't concerned about this and would rather test manually, the idea is to boot to a DOS command line and experiment with setting your PC to about 11:58pm on December 31, 1999, and letting it run into 2000. If it does, turn the computer off, wait five minutes and then restart it to see whether it retains the post-2000 date and time. Also, test switching your PC off with the time set at about 11:58pm on December 31, 1999, then switch it back on about four minutes later to see whether the BIOS correctly interprets the new century. Don't be disappointed if the clock rolls over to 2000 without difficulty — but it is quite dramatic to see a clock calmly jumping from 1999 to 1900 or 1980 at the stroke of midnight!

Hardware with non-compliant BIOSes needs either to be upgraded, given a helping hand by software, or completely replaced.

If you can find out the brand of your motherboard, go to the support section of the company's Web site and look for a downloadable 'flash' BIOS upgrade which will overwrite your existing BIOS with a Y2K-compliant one. Note that flashing a BIOS is a dangerous process that *must* be done from a simple DOS command line with absolutely nothing else loaded into the PC's memory (start your computer with a suitable boot disk or press Shift-F5 when Windows 95/98 starts up). The flash upgrade you are using *must* be the one intended for your motherboard. Follow the manufacturer's instructions to the letter and pray that there isn't a power cut during the flash

process, or you may end up wrecking your motherboard (it's usually too difficult to find replacements for ruined ROM chips).

If you'd rather not risk this or there isn't a suitable upgrade for your BIOS, you need to install a piece of software that assists the BIOS to operate in the correct century. Generally this program will take the form of a TSR (terminate and stay resident) program which runs from your CONFIG.SYS or AUTOEXEC.BAT startup files. There are many different types, each claiming superiority, and apart from Viasoft's freebie on the Pocketbook CD, they all cost money. Some are included in Y2K testing software, and some are standalone products (they are very expensive for tiny, simple programs).

The thing to note with a software patch for the BIOS is that it *must* remain on the PC and be reinstalled if deleted. If it isn't loaded, a non-compliant BIOS will go back to doing its own thing and create havoc.

Very old (in computer terms) hardware, such as that dating from the 1980s, will often have more severe BIOS problems that can't be fixed at all. For example, such a BIOS may only be able to cope with the period 1980 to 1999. In that case, all that can be done is to replace the machine, or perhaps add a card or dongle (see page 62).

OPERATING SYSTEM

Even if your BIOS and RTC are as ready as they will ever be, everything still depends on the operating system (OS) you use with them. If your OS can't take the heat, then all the work you did to test and fix your BIOS will be irrelevant.

Every operating system varies in its level of Y2K compliance. No operating system is immune to date-related problems if it is tasked with running third-party software (and that's what OSes are for), but some are compliant in themselves.

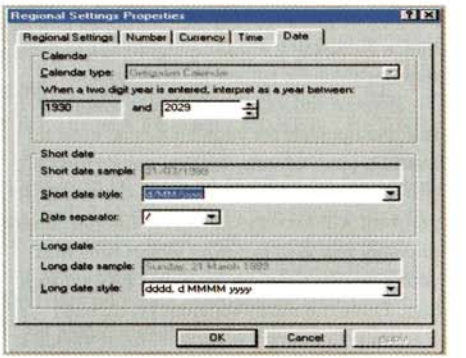
Microsoft gets no marks at all. No version of MS-DOS is really Y2K compliant, because that operating system was designed for machines which could only handle the period January 4, 1980, to the end of 1999. Commands like DIR can only show two-digit years in their output. Versions 5.0 and above can handle four-digit years and will function post-1999, but the OS still presumes two-digit years are in the 20th century. Microsoft isn't going to help anyone with this. Other versions of DOS, such as Caldera's DR-DOS and IBM's PC-DOS, are only a little better. Software which runs on DOS systems that will be active after 1999 must also be checked.

Microsoft OSes

Windows 3.1x and its predecessors also aren't Y2K compliant, though a modified version of File Manager is available to show years in four digits. Windows 3.1x still works out the current date by adding a number to January 4, 1980.

Windows 95, contrary to Microsoft's hype, simply sits on top of DOS and is also vulnerable to Y2K. It is imperative you apply a patch from Microsoft. One such patch is included on the Pocketbook CD. Apart from the OSR 2.5 revision (which is similar to Windows 98), the operating system's clock settings will only handle two-digit years in the range 80 to 99,

Flash BIOS upgrades can be dangerous



Remember to expand the short date format in Windows 95/98

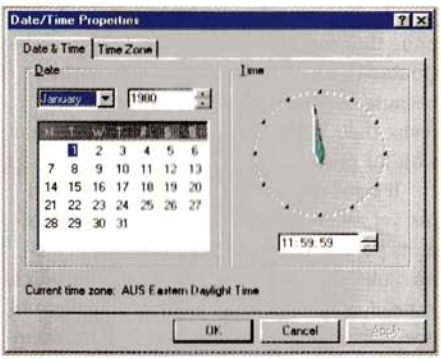
and regardless of your system clock's compliance it will treat 2000 as 1980 (if you don't believe Microsoft could have overlooked this, just try setting the date outside those ranges). This is evidence that simple BIOS patches are not 'complete' Y2K solutions, although this is how some companies promote them. You *must* address operating system issues yourself.

Note that the release version of Windows 95 treats the years 1980 to 1989 as 2080 to 2089 — interesting, but not very useful unless you plan to avoid using your PC for the next 80 years. (If by any chance you happen to be still running a public beta version of Windows 95, the years 80 to 99 will be treated as 1980 to 1999). The solution to Windows 95's Y2K freakishness is to either add a patch from Microsoft or upgrade to Windows 98.

Windows 98 was supposed to be ready for Y2K, but it is not, and it took external pressure to get Microsoft to admit this. It is "compliant with minor issues", according to its makers. In some situations, installed as originally released, the operating system *may* not roll over correctly into 2000 (for example, if it is booted during the date rollover). Again, you must apply a patch from Microsoft to address this. You should also apply a patch for Microsoft's Java Virtual Machine (used with Internet Explorer 4.0) to correct separate issues.

There are also a few problems with minor components of Windows 98. The files mswallet.exe, dialer.exe, comctl32.dll, timedate.cpl, docprop.dll and a few others all have separate Y2K issues and must be upgraded in addition to the installation of the main Windows 98 patch (look on the Pocketbook CD and check Windows Update).

Windows NT 3.5x is not Y2K compliant and requires a few significant fixes, while Windows NT 4.0 comes closer, but is not completely ready for 2000. In both cases, the latest service packs from Microsoft are prereq-



Windows 95 reverts to 1980 instead of 2000

Get the latest Y2K information from software company home pages

uisites for compliance (and must be installed if Microsoft is going to continue supporting you on Y2K-related problems with NT).

Current Windows 2000 betas appear to be OK . . . but you can't be sure. Windows CE is also "compliant with minor issues" and Microsoft's Web site hosts the "minor" patches.

One thing you must watch in 32-bit versions of Microsoft Windows is that the year display value for the short date format in the Regional or International settings (in Control Panel) is set to yyyy, not just yy. This doesn't cause problems directly if left unchanged, but it's a good policy to make the change just in case (and for aesthetic reasons — after this crisis we're going to baulk at shortening the date to 00 or 01!).

Other OSes

What about non-Microsoft operating systems? Well, if you're lucky enough to run an Apple Mac operating system, a modern Unix variant (like Linux, BSD or Solaris), or something out of left field like BeOS, you don't have half as much to worry about. Mac OS has a couple of small, recently uncovered Y2K issues with system settings (see Apple's Web site for the latest advice), but the risks are low. Unix users *must* make sure they have the latest version of their operating system, as early versions of OSes like IBM's AIX and Sun's Solaris are definitely not ready for Y2K.

Thanks to the dedication of the community behind its development, Linux has been Y2K-ready for a long time. The same goes for a few modern Unix-based apps such as the ubiquitous Apache Web server, for which Version 1.2 or earlier are not compliant, but 1.3 or later are fine.

As usual, there are potential problems with third-party software and the machine's BIOS that could bring all your confidence crashing down, but with Unix variants (and BeOS) for example, the critical date is actually

a few decades away — January, 19, 2038, at 3:14am. This is due to the way these operating systems calculate dates, by working upwards from January 1, 1970, and treating time as a 32-bit variable, which means there are only 2,147,483,647 seconds to work with (2038 is when these will all be used up). No doubt people will still be using a descendent of Unix in that year, so some kind of solution will be needed long before the event; it will probably just involve turning the 32-bit value into a 64-bit value (which would give Unix another 292,271,023,017 years before the next crisis).

So, for those of you caught up in the Microsoft Y2K quagmire, perhaps it's time to move to Linux or buy a Mac!

SOFTWARE ISSUES

It is expected that a great deal of software, particularly custom-written software, will not be compliant by 2000. But ensuring software is compliant is a vital task in your Y2K preparations once your firmware and operating system have been dealt with.

Aside from problems with internal calculations, older software may simply ignore the century when saving creation and modification dates — a file might be date-stamped 01-Jan-00, instead of 01-Jan-2000. If the saving application or related software can't guess the date correctly, you could have problems in areas such as date-sorted search results and figuring out which file is the most recent save (programs may even overwrite or save to the wrong file).

The first place to start, once again, is the manufacturer's Web site or support line. If you don't know where to look, call the company that supplied the software to you. Double-check as the new year approaches, because companies have been known to revise their Y2K compliance information and wait for people to notice.

If a software or hardware company has not yet made statements on the compliance of its products and made compliance documentation available, supplied upgrade patches or at least recommended that you move to a whole new version, then you should strongly consider switching to new software for that reason alone.

Unfortunately, even if we had two Pocketbooks, it would still be impossible to cover all the possible Y2K problems of all the software on the market, so we'll concentrate here on office productivity suites and other commonly used applications.

Microsoft products

Again, we'll start with Microsoft, partly because its products are in such widespread use, and partly because its Y2K record is a little embarrassing. As a general rule, Microsoft products should be compliant if you have installed the most recent Service Pack (see the cover CD and Microsoft's Web site under 'Year 2000', where all the company's products are indexed with Y2K findings) and your operating system and hardware are up-to-date.

The 32-bit (Windows NT, 95, 98) versions of Excel, Word, Access, and so on, are all either compliant or capable of being made compliant. The older the application, the more alarming the potential problems. For

example, Word 5.0 for DOS has a tendency to freeze due to invalid file dates. It is not a good idea to continue relying on 16-bit (Windows 3.x) or DOS versions, because sooner or later you will run into two-digit years. But just because it's a 32-bit program doesn't say anything about Y2K compliancy. For example, FrontPage 1.1, released for Windows 95 in 1996, cannot handle date and time calculations past 2000. And just because it's 16-bit doesn't mean it can't be made compliant (presuming you have the sense to run it on a compliant operating system) — Y2K patches are available for Microsoft Mail, even though the company is intent on pushing its current Outlook mail program.

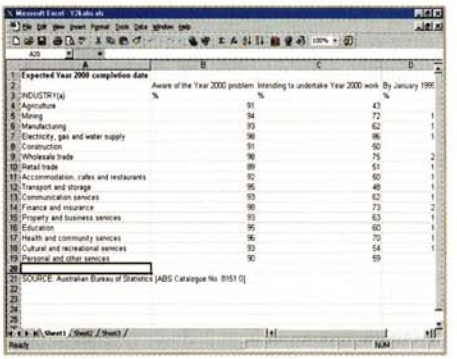
Older versions of Microsoft applications for Mac (such as Excel 5.x) are also problematic — remember, Macs may be compliant at hardware and operating system levels, but are just as vulnerable to software and data problems. Excel 98 for Mac is compliant.

Other products

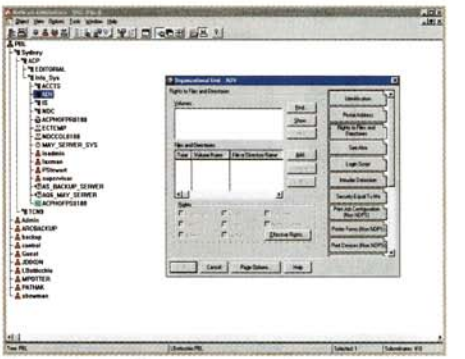
Microsoft's competitors in the productivity software market are in similar strife. Elements of Corel's WordPerfect Suite have minor Y2K problems prior to Version 8.0, as do pre-Corel versions of the WordPerfect word processor. Lotus eSuite is not compliant without an upgrade, due to known problems with the SunSoft Java Developer Kit on which it depends. The same goes for software from other companies such as Netscape, which is only guaranteeing shipping 4.x products and above, due to Y2K issues with previous SunSoft Java Virtual Machines.

But Lotus likes to say that all components of its SmartSuite are "Year 2000 ready". What it should really be saying is that up until SmartSuite 97, all two-digit years were assumed to be in the 20th century (four-digit years were recognised, but had to be deliberately entered) and now Lotus has begun to rely on pivot dates.

In SmartSuite 97 all dates with a year between 00 and 49 are assumed to be 20xx. Dates between 50 and 99 are assumed to fall in the 20th century. In its Millennium Edition of SmartSuite (the name at least shows confidence), Lotus has switched to a radical 80/20 split. Two-digit years up to 80 and prior to the current year are assumed to be in the current or previous century. So, if the current year is 98 and the user inputs a date with a year of 25, 1925 will actually be entered. But if the user enters 05, the program will see 2005. Likewise, if the current year is 2010 and the user enters 05, the program will still see 2005.



There will be refences to two-digit years in your data



Network operating systems are also susceptible to Y2K

But the number 25 produces the year 2025.

One Lotus product which definitely isn't compliant is cc:Mail. If you still rely on it for your email, you should upgrade to another, backward-compatible email package. Lotus Notes has always been Y2K compliant, but as it is almost a programming environment in itself, it's possible for Notes developers to have introduced date-related problems when they created databases and applets. The original authors of such code are the best people to sort out this kind of situation, but they may not be available to do so. There are new tools on the market that help with Notes remediation tasks.

For the same reasons, any modified or heavily customisable applications are vulnerable to introduced Y2K errors. Macros written for office productivity applications, scripts written for Unix environments, batch files, and so on, are all a source of possible problems.

Obviously, a problem for larger or more specialised environments will be purpose-written software that no-one else uses, and which only a programmer can really fix. It is said that the Social Security Agency in the US had to revise about 80% of its software, because it was reliant on applications and customisations developed inhouse.

It isn't really necessary to go into great detail here about how programmers fix software (if you're doing it yourself, you will need a much more comprehensive guide). Suffice to say that it involves updating the inherent methods used by the program to handle timing and date-reliant operations, as well as how it displays and records dates. Various companies produce automated programmers' tools that speed up the process, while MFX research claims MFX 2000 allows any non-programmer to change bytecode automatically, making software compliant without the need for source code.

Problems with personal finance packages have been widely reported, as these tend to be much more date-reliant than other appli-

cations such as word processors. The best advice is to keep up-to-date to reduce your risks — for instance, older copies of Quicken (such as Versions 1 to 4 for DOS) simply won't work in 2000.

DATA PROBLEMS

Tracking down and altering problems in data created over many years is often cited as the most time-consuming part of Y2K remediation. Apart from the tedious process of adding an extra two digits to all dates in historical data so that it doesn't confuse present operations, it's also necessary to ensure you don't install any new software that introduces new errors and that data exchange with third parties doesn't put compliant environments back into the danger zone. This includes word processing files, spreadsheets, database files and proprietary formats.

Note that spreadsheets are often the primary data danger on PCs. Many businesses, big and small, maintain complex Excel files that contain critical day-to-day operations and forecasting information. Programs like Datespy (http://www.datespy.com), Norton 2000 and Viasoft's OnMark 2000 offer specialised assistance with Excel.

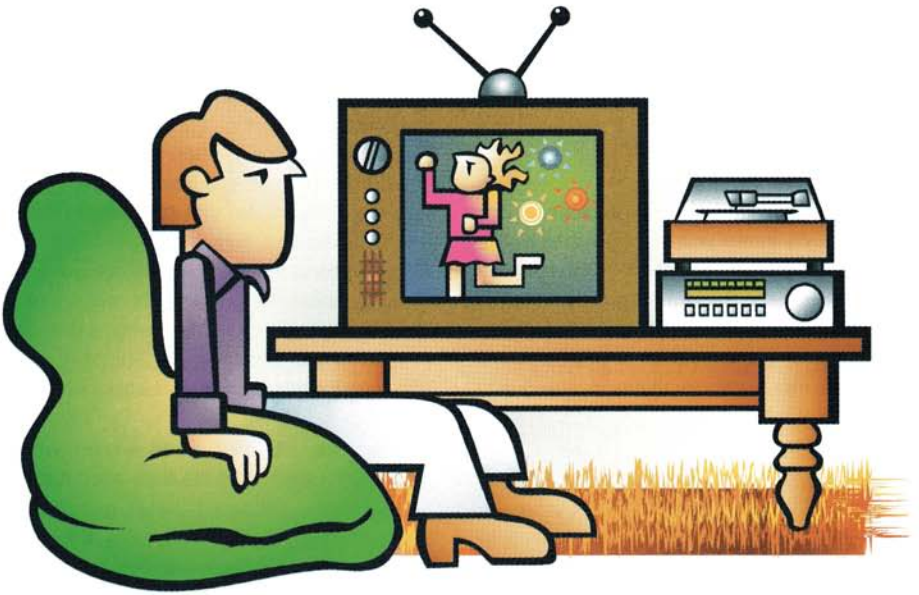
The reason it's important to make data compliant is that your hardware, software and operating systems are only as good as the information they receive. Calculations and computer operations are likely to run into problems if they ever refer to non-compliant information (and they eventually will). Therefore, Y2K data repair is an ongoing process, which will continue well into the next century.

An alternative solution to all this repetitive data correction is to use software that supports 'windowing' or 'pivot dates' (we'll use the latter term here to avoid confusion with graphical windows in operating systems) to define the space of 100 years within which your computer will place dates with two-digit years. For example, Microsoft Windows 98 has a pivot date capability that defaults to 1930 to 2029; a two-digit year of 30 or higher is automatically assumed to be 1930 or higher, whereas a two-digit date between 00 and 29 inclusive is assumed to be between 2000 and 2029 (inclusive).

> Y2K DATA REPAIR IS AN ONGOING PROCESS, WHICH WILL CONTINUE WELL INTO THE NEXT CENTURY

The fact that the latest operating systems ship with such a feature is of great assistance to anyone undertaking last-minute Y2K work — if you can't fix it, you can compensate for it. Providing your pivot dates are set sensibly (and perhaps moved forward one year, every year) the most common dates that your computer will encounter are safely catered for.

Pivot dates are also built into many programs, but they vary across programs, so it's better to let the operating system handle things if possible. Otherwise, 1950 in one



program could be 2050 in another. Pivot date systems are far from perfect. It would be far better to move completely to four-digit years and be done with the trouble.

NETWORK COMPLIANCE

Checking non-PC hardware, such as network components (such as hubs and routers and purpose-built servers), is more difficult. Often these are machines that have complex and date-reliant firmware but can't run automated tests themselves. The idea is to do it across the network and request time information from the component.

But the first place you should start is the manufacturer's Web site. All network hardware manufacturers have placed online Y2K compliance information pertaining to their equipment. Generally the issues will be described and categorised according to severity, and there may be downloadable firmware upgrades or advice on workarounds. Be sure you match the advice or patch with the precise model and version for which it is intended.

The same goes for dedicated network operating systems (NOSes). You should check whether older versions of software such as Novell NetWare, Banyan Vines and Artisoft LANtastic, are 2000 compliant, and if they need patches or complete upgrades. Each

company has addressed Y2K; if you want to stick with the same product, you just have to make sure you have an up-to-date version.

EMBEDDED SYSTEMS COMPLIANCE

As discussed earlier (see page 34), some embedded systems can be made compliant by overwriting their instruction sets. Many cannot, so the usual practice is to completely replace either the affected chips or systems. This is generally a problem for business and government — contrary to some media advice, it isn't necessary to replace all your household equipment. Some household problems may become evident with non-computerised electronic equipment such as VCRs. If your VCR is not Y2K compliant and you want to continue using it to perform timed recording or playback, you will need to implement a workaround.

One that has been repeatedly suggested is to turn the date back to 1972, during which the days of the week fell on the same calendar dates as they do in 2000. This is a bizarre solution and its effectiveness would vary depending on your particular situation — and many VCRs don't need to know the year anyway.

Other home and office equipment such as fax machines, security systems, thermostats and medical equipment may have date-sensitive microchips. One telltale sign is if the product has an LCD. Usually such equipment will work but will give the wrong date, so you might just have to grin and bear it, unless displaying the correct date is absolutely vital (such as on a fax machine used to transmit legal documents).

COMPLIANCE STATEMENTS

When replacing or upgrading any equipment, it's critically important to receive in writing a statement of its Y2K compliance. Beware, however, of varying definitions of compliance.

For example, most companies use similar definitions to that of Lotus: "Lotus considers a product Year 2000 ready if the product, when used in accordance with its associated documentation, is capable of correctly processing, providing and/or receiving date data within and between the 20th and 21st centuries, provided that all other products (for example, hardware, software and firmware) used with the product, properly exchange accurate date data with it."

Using this definition of compliance, Lotus is able to say that because older versions of its 1-2-3 spreadsheet are able to work with four-digit years, they are compliant. But they do not use four-digit years by default and there is a risk of data problems down the line. The statement also allows for pivot dates to be used as a compliance measure, and does not guarantee compliancy with dates beyond the new century.

As always, public relations, marketing and corporate image priorities can get in the way of any company disclosing the plain truth. At the end of the day you must be satisfied yourself that it is possible to reliably use what you buy in a manner that remains Y2K compliant. If you have it in writing, and you have done everything required of you, then you should be on solid legal ground if anything goes awry.

The apps

REPAIR SOFTWARE

PC software tools to help fix Y2K first appeared on the market in mid-1998, which already seemed a little late. New products are still appearing (keep an eye out for the latest reviews) and the established products continue to evolve.

Generally they fall into two categories: those which test and fix your computer's BIOS; and those which also examine your operating system, software and data. The latter, fully featured programs are now the market leaders, and some of the newer products are actually licensed versions of these, sold under different brand names.

Here's our take on a few of the products you're likely to find in the shops. Unfortunately, no tool is perfect, but while each does something better than the others, the top-notch offerings do cover all the most important areas in one way or another. If you want to tackle Y2K this way, your best bet is to pick a good application and run with it.

Viasoft OnMark 2000

This is also resold as Symantec's popular Norton 2000. When Y2K professionals speak of PC analysis and repair tool companies, they generally name Viasoft and Greenwich Mean Time (see below) — indeed, each firm sees the other as its main competitor.

THE TOP-NOTCH [TOOLS] DO COVER ALL THE MOST IMPORTANT AREAS IN ONE WAY OR ANOTHER

Viasoft's background is in high-end mainframe software, and OnMark 2000 is actually a suite of applications for examining, fixing and retesting PC hardware, software and data. The components of the suite can be purchased either separately as the need arises, or all in one hit. Businesses will require more parts of OnMark than home users.

The most basic element of Viasoft's package is BIOS Test & Fix, which can install a program that compensates for non-compliant firmware. It originally shipped with Assess, the company's data scanning tool, but has since been made available over the Internet as a free download, technically making competitors' BIOS-only products obsolete. This is because the company wishes to make the point that BIOS problems only represent about 10% of the Y2K risk on PCs; software and data are a much bigger issue. Viasoft's BIOS Test & Fix is included on the Y2K Emergency Pocketbook cover CD.

Norton 2000

Symantec
Phone (02) 9850 1005
Online http://www.symantec.com.au
Price $79

Survey is the part of OnMark 2000 which examines the software on computers across a network and reports its Y2K compliance. It's really just an inventory tool relating only to software. Viasoft's WebCenter is a more complete inventory and analysis tool run from a Web site (it's designed for companies running an intranet).

As mentioned above, the Assess scanning tool is perhaps the core of OnMark 2000. It's the part that looks at your data files (everything from text files to complex spreadsheets) and finds examples of possible two-digit date problems. Its output can be quite verbose and it does tend to see problems in some non-date related data, but you're better safe than sorry.

There's quite a lot of hard work involved in going through scanned data and referring back to the Assess report to make changes. But that's what Y2K repair is like — the more data you wish to make fully compliant (rather than using a pivot date to fudge it), the more elbow grease you'll need. Assess at least simplifies the hunt, and a server-based version is available for larger networks.

The other parts of OnMark are all 'Workbenches' — modules which are worth adding in particular circumstances. Most users will not need these, and they are very expensive. If you need specific Y2K help on a large scale with data produced by Lotus Notes (which is compliant in itself, but can suffer problems with inhouse adaptations and data), Microsoft Excel and Microsoft Access, or code produced in C/C++, Visual Basic, PowerBuilder or Unix shell scripting language, there is a relevant Workbench available.

Greenwich Mean Time Check 2000

Greenwich Mean Time (GMT) is a British/South African firm founded by Y2K guru Karl Feilder, who travels the world commentating on Y2K issues, advising governments and business on their preparations. His 'five-level' theory of Y2K (see page 30) forms the core of Check 2000's approach to the problem.

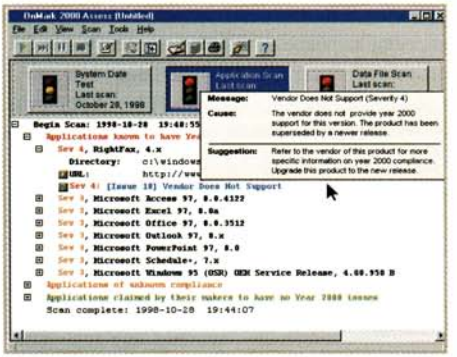
A basic version of Check 2000 has been licensed to IMSI in the US as part of Year 2000 Now, but the licensing agreement does not currently allow it to be sold in Australia. That may change soon, however, and some copies of

OnMark 2000

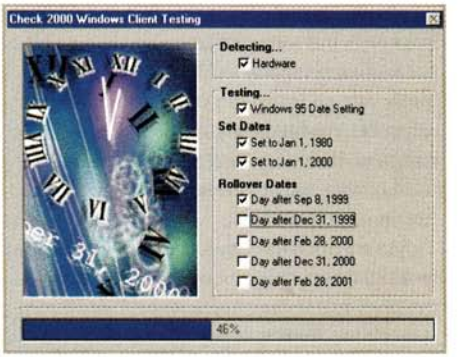
Viasoft
Phone (02) 9460 0411
Online http://onmark.viasoft.com
Price $75 single user
(also multiple licence pricing)

Check 2000

Greenwich Mean Time
Phone (02) 9900 5360
Online http://www.gmt-2000.com
Price $79 single user (also multiple licence pricing)



OnMark 2000 checks your software compliance



Check 2000 tests various BIOS dates

> GREENWICH MEAN TIME IS A MORE USER-FRIENDLY APPLICATION WHICH TAKES A DIFFERENT APPROACH TO Y2K

Year 2000 Now were imported and sold locally in early 1999. The original Check 2000 comes in several guises: a single-licence personal edition and its Gold sister, sporting added capabilities (both priced under $100); the five-licence small business edition (around $350); and the full client/server version for network administrators (sold on a per-licence basis).

Greenwich Mean Time is a more user-friendly application which takes a different approach to Y2K. Naturally enough it begins by taking a look at the BIOS of a PC and testing its capacity for dates such as 9/9/99, the 99 to 00 rollover, the leap year in 2000, and so on. If your PC is not compliant, you click a BIOS Fix button and software that compensates for the date rollover will be run from your AUTOEXEC.BAT file when your PC starts up.

Instead of scanning the software on your hard disk from scratch, Check 2000 refers back to a database created by GMT's testing labs in various countries. Y2K compliance information for about 20,000 applications is held in the program's database, so it's likely that the most important applications on any hard drive will be covered. The software recommends possible courses of action based on what it finds.

Double-clicking on a listed application brings up deeper information from GMT's help file. This is where Check 2000 comes into its own — the program ships with some of the best Y2K documentation in the business. Everything is written in plain English with no hype or waffle.

The data scanning stage of Check 2000 allows customisation — it asks the user to

Prove It 2000 checks your RTC

select which file types and date formats to look for. Apart from everyday files such as those created by spreadsheets and databases, the program can also handle several macro and scripting languages. The downside is that the data tool is noticeably sluggish — some people find this very annoying, and it could probably have been implemented better.

As with Viasoft's Assess tool, Check 2000 is meant to be re-run every now and again as 2000 approaches to check for continued compliance.

MFX 2000

Formerly known as Millennium Master, MFX 2000 is an Australian product invented by MFX Research's Bruce Parker, programmer of the venerable Windows crash protection program RamGate. Its chief claim to fame is a set of automatic correction algorithms which enable the software to find and fix Y2K problems in software and data. MFX claims its product can actually go into commercial software and make the bytecode Y2K compliant, without requiring any source code or changing file checksums (so your virus scanners won't know the difference).

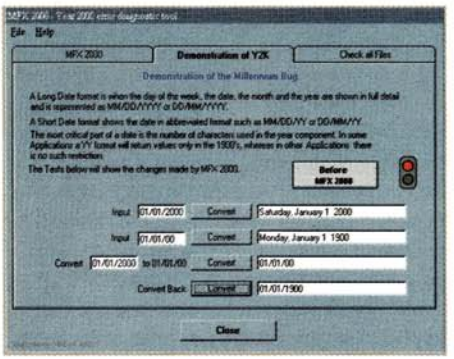
The checksums of data files do increase, but that's just a natural progression of the fact that they contain an extra two bytes of data for each two-digit year that is changed to four. However, changes made by MFX 2000 must be consistent across an entire network to ensure data integrity. The company's most controversial claim is that this PC-only application can be used to correct code on mainframe drives that are mapped to Windows using Samba.

Trusting such a process is a major leap of faith for many companies, but there is plenty of interest in MFX, and despite some shaky financial backing early on, the company's fortunes seem to have picked up. Competitors allege that automatic repair is impossible; for example, how is the product to know if 12/99 refers to December 1999 or 'Unit 12, 99 Main Street'? But the MFX algorithms make several passes over code and data, apparently searching for a match against 40,183 specific

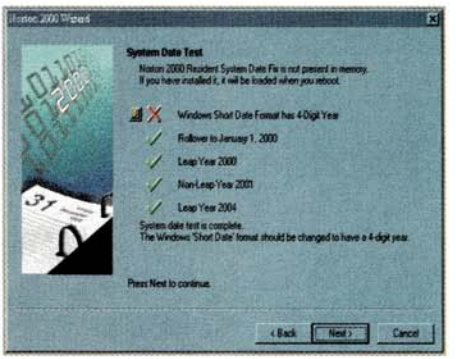
MFX Research

MFX 2000

Phone (02) 9440 0200
Online http://www.mfxr.com
Price $95 audit tool
$250 audit and correction tool (also multiple licence pricing)



MFX 2000 shows how your PC handles dates



Norton 2000 checks Windows' short date format

date formats. Unfortunately it's impossible to verify what the program actually does, as the algorithms are entirely secret.

MFX 2000 is not quite as pretty as its rivals, taking a purely functional approach. First, it scans everything on your hard disk to find out what's there. Then it allows you to examine these results and select individual applications and files to alter. Once you give the approval to make permanent changes, program code flashes by impressively as MFX 2000 does its work.

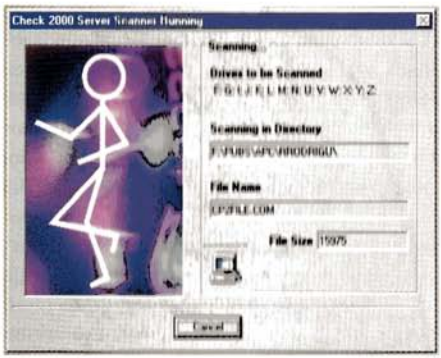
Prove It 2000

Big in the UK, Prove It 2000 is a little newer to the Australian marketplace than some of its rivals. A single-user version is available which offers an eight-point BIOS check and fix process; but the major offering is a pricey application for businesses that takes a server-based approach to managing Y2K issues across many PCs simultaneously. It's

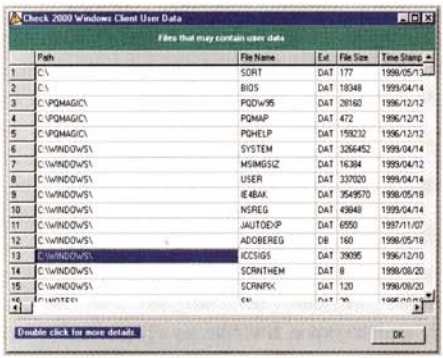
PROVE IT 2000 TAKES A SERVER-BASED APPROACH TO MANAGING Y2K ISSUES ACROSS MANY PCS

equal parts inventory tool, BIOS testing/patching utility, and front-end for managing software problems. The locations of patches that must be added to various PCs on a network can be entered into the application so that the administrator can select a PC and send the patch there.

When it comes to the actual 99 to 00 rollover, Prove It's BIOS testing routine is a little more complicated and thorough than the others, requiring the user to perform both power-on and power-off testing (the power must stay off for a certain period, after which the PC reboots and the user is told what happened. Many PCs would work fine if they remain switched off during the rollover, but would revert to 1900 or 1980 if



Check 2000 client/server examines the network



Data scanning reduces the most tedious work

they were left on. Obviously this is a serious problem for server administrators (it wouldn't be much fun missing out on new year celebrations because you have to reboot some computers).

In comparison to the glitzier GMT and Viasoft products, Prove It offers significant benefits to system administrators in managing Y2K problems, and it has perhaps the simplest interface for applying BIOS fixes remotely; but the competition offers more thorough facilities for actually scanning software and data.

Prove it 2000

Prove It 2000
Phone (02) 9388 3126
Online http://www.proveit.com.au
Price $129 single licence

And more . . .

Not to be outdone by Symantec's Viasoft clone, popular utility vendor McAfee has released its own Y2K package called 2000 Toolbox. It contains yet another BIOS fix and more scanning tools for software and data. The BIOS testing and repair component is also available in McAfee's Nuts & Bolts system optimiser package, and McAfee's implementation seems quite solid.

Vet 2000 is another recent release (from the same company as the successful Vet virus scanning software), but its producers have struck an agreement with a service station company for wider distribution. The basic application costs $10 and the company freely admits it's nothing more than a BIOS testing tool. If it finds a problem, the user is asked to send away for a BIOS fix costing $99. The program is well-presented, but it's hard to justify this price tag, given that Viasoft's equivalent

has been available free since late 1998. Competitors such as Millennium Buster and Centennial 2000, also priced at around $99, offer nothing new, although Centennial 2000 is also available in a client/server edition for deploying BIOS patches across a network.

There are many more Y2K testing and repair tools on our Pocketbook CD and on the Internet. If we haven't included your ideal solution on the CD, try starting at a site like http://www.year2000.com, run by the original Y2K expert, Peter de Jager. And beware of opportunists trying to cash in with 'magic bullet' solutions; a few of the lesser-known applications available for download appear to be shonky pieces of programming which may do more harm to your system than good.

BEWARE OF OPPORTUNISTS TRYING TO CASH IN WITH 'MAGIC BULLET' SOLUTIONS

There is another method of fixing BIOS and it's more permanent than simply installing a small piece of software. ISA cards and dongles that override your current BIOS with compliant instructions have recently become available. These are especially useful if your PC is particularly old (too old for a simple software fix) and cannot be replaced, but they do tend to be on the expensive side. At the time of writing, several PC hardware shops and mail-order services around Australia stocked them for $250 to $300.

Vet 2000

Computer Associates
Phone 1300 364 750
Online http://www.vet.com.au
Price $9.95 test, $99 fix

McAfee 2000 Toolbox

Network Associates
Phone (02) 9437 5866
Online http://www.mcafee.com
Price $69.95

Testing

TAKE THE TEST

For the most part, testing the results of your Y2K preparations involves simply completing all remediation work as quickly as possible, then carrying on your normal activities to see if anything goes wrong. If you experience some problems in pre-2000 testing, be grateful — it's considered much better to have something fail now than post-2000, because it identifies something that can be fixed. Shareholders of big firms love it when the company's computers fail in Y2K testing because it means one more problem has been tracked down and eliminated!

Proper testing also involves exercises in creating possible risky situations. As with anything computer-related, it's important to minimise interference with something that works. Even if it's behind the times, a system that gets the job done on time and on budget is worth much more than one that's up-to-date but unstable, so it isn't worth rushing headlong into a simulated future with all systems running full speed ahead . . . unless perhaps you're messing about with your own PC out of interest.

Large businesses should be able to afford some experimentation with redundant systems, so that it's possible to duplicate core practices and place them forwards or backwards in time to see what happens, without interrupting current work. If you do this, make sure your testbed is as close to identical to the real system as possible.

Large businesses use automated testing packages which can simulate real users around the clock over coming months. For the rest of us, it's probably a matter of becoming ever more adventurous as we give things a try.

Obviously, the first test you might think of doing after all the work is done is placing everything a year or so into the future to see what happens. But this is inadvisable as a first step in most situations — if any calculations involving dates take place while you do this, the chances are they will write odd pieces of data just where you least expect them. Even worse, you might forget that you changed the system date and created new files with completely incorrect date stamps or content.

Take one step at a time. Introduce a few deadlines or due dates with new millennium dates; create some cost projections into

IT'S FAIRLY EASY TO REINTRODUCE OLD PROBLEMS OR CAUSE NEW ONES, SO STAY ON YOUR GUARD DURING THE TESTING PERIOD

2000; make some deliberate date-related errors and see if you can cause a little bit of havoc. If you're really confident, you might then play with the system date in controlled conditions — watch the machine roll over to 2000 and have another play.

Note that it's fairly easy to reintroduce old problems or cause new ones, so stay on your guard during the testing period. For example, if a hard disk must be reformatted it will lose any software patches and BIOS fixes you've installed. Even the reinstallation of a single program without the addition of a necessary Y2K patch could create a problem. Meanwhile, until 2000, regular rescanning of data should be approached in much the same way as a regular disk defragmentation or virus check.

Interrupting your daily work or that of your colleagues is sometimes inevitable. Sometimes the person who created something (data or a piece of software) is the only person who knows how it really works, and that person must therefore be involved in repairing and testing. If important work must be interrupted, consider doing so on a planned basis where more than one thing is tested at a specific time — perhaps at a regular time each week, so that it becomes less intrusive and part of a routine.

Hopefully, the parties with which you exchange data will also be carrying out testing. Depending on your relationship, it may be worth arranging some kind of joint test to see if the changes you've made to your respective equipment are compatible. For example, if you have made the effort to change every date and update your software, but someone supplying data to you is relying on pivot dates and workarounds, things may go awry.

Checkpoint — Testing procedure

- **After completing remediation, carry on normal activities**
- **Be careful to avoid introducing risky situations**
- **Play around with a few new dates at a time**
- **Test on a regular, routine basis to minimise chances of disruption to workflow**

y2k
Contingency planning

Chicken Licken?

WILL THE SKY REALLY FALL after the Y2K bug hits us? Is it the end of life as we know it? Or will we simply suffer minor inconveniences? You can readily find experts predicting each of these extremes, and just about everything in between. Perhaps the most worrying part is that we don't know for sure, and can't know for sure until it's too late.

In Australia, more so than most countries, you *can* probably discount the worst doomsday predictions. For example, our power grids have fairly low levels of reliance on IT, so long-term disruptions are pretty unlikely. Even if they do occur, it'll be the middle of the summer holidays: the weather will be warm and many industries will be closed. Compared to some Northern Hemisphere locations where it will most likely be snowing, Australia is probably a good place to be in January 2000. However, it would be unrealistic to expect that Y2K won't cause any problems at all here. You probably don't need to build a bunker, but you certainly need to take a few precautions.

So, how do you prepare for problems when you don't know what they will be? A large part of Y2K planning is just general contingency planning. If tomorrow morning the phone or power service were to fail or a water pipe burst, what would you do? If a supplier closes its doors or a big customer can't pay its bill, how would you handle that? What if your computer's hard disk crashes? These are all things that might happen in January 2000, but they could also go wrong tomorrow. If you could handle them now you are well on the way to being able to handle them in January — although there are a few extra complications.

There are two main strategies for contingency planning. Comprehensive contingency planning is the strategy of choice when you are pretty much prepared for Y2K. This is 'belt and braces' safety where you've dealt with everything you can think of, but are preparing for the unforeseen. Triage contingency planning is the strategy you adopt when it becomes apparent that you aren't going to make it. It is primarily concerned with harm minimisation rather than prevention. We will concentrate here on comprehensive contingency planning, as triage, at this stage, is more relevant to large businesses. We will, however, briefly discuss triage near the end of this section.

BACKING UP

Will your computer implode on January 1 and cease to function? Probably not. However, a rogue program *could* damage important data, or a power spike *could* fry your hard disk. How quickly could you recover if this happened? This probably depends on how good your backups are.

There is a great choice of backup devices, from the humble floppy to the $260,000 DAT library. For most people, neither of those extremes will do. The choice of backup device is mostly driven by the amount of data you need to back up. You don't want to spend $10,000 or more on a 25G DAT drive just to back up 2M of documents, but neither do you want to spend all night swapping disks as you back up your 20G database onto 100M Zip disks.

There are two ways to determine what to back up. The simplest is to back up everything on your hard drive. This requires little setup and less maintenance, but it means that you need a backup device big enough to store everything. An alternative is to just back up data files. This usually works well for home and small business computers.

Applications, such as Microsoft Office, can use up large amounts of hard disk space. You don't generally need to back these up, because you can reinstall them from the original install disks. This leads to quicker, cheaper backups, but it means that you must be careful to add any new work to the backup set.

The lowest branch on the backup tree is the floppy disk. Not too many years ago, a floppy drive cost a couple of hundred dollars and disks cost several dollars each. Today you can buy a floppy drive for $30 or $40, and the floppy disks cost only a few cents each. The trade-off has been reliability. Today's drives and disks are not reliable and you should think twice before using them for critical backups. If you really must use them, make two or

three copies of each backup so that you can recover if one or more disks has bad sectors.

Removable disks are a new and popular option for backups. Zip, LS-120, Jaz and others provide fairly simple, cost-effective solutions. Because these devices act like normal disks, you can simply copy data onto the disk, rather than using proprietary software with proprietary data formats. These devices are not usually suitable for backing up a whole hard disk because they don't have the capacity, but they make a good option for backing up selected data.

Tape drives can be a good choice for backing up a whole hard disk, or even an entire network. Travan drives offer fairly high capacity at a budget price, the largest drives holding 4G on one tape (or up to 8G if you compress the data). These drives can be a little slow, but they remain good value for money. DAT drives are faster and are available in very high-capacity configurations. Multi-DAT libraries are suitable for backing up even the largest networks. DAT drives cost more, but they provide a comprehensive solution for business backup. The biggest downside to tape backups (other than cost) is the need to use archive software. On Unix systems there is standard software such as tar and dd that use well documented formats. However most commercial backup software uses proprietary formats which cannot be read by other packages. If you use such software to get around Y2K, you had better get a cast-iron guarantee from the supplier that it will function next year. January 2000 is *not* the time to discover that your backup software will not do a restore!

CD-R and CD-RW drives are also becoming popular as backup devices. The prices of drives and media have dropped sharply over the last year or so, making them a realistic choice for home or small business. With a capacity of 650M they are too small to completely back up many computers, but remain a good choice for selective backups of data. Although you use special software to write to CD-R and CD-RW disks, you can read them in any standard CD-ROM drive — so you don't have to sweat about the compliance of your backup software.

TODAY'S DRIVES AND DISKS ARE NOT RELIABLE AND YOU SHOULD THINK TWICE BEFORE USING THEM FOR CRITICAL BACKUPS

BUSINESS CONTINGENCY PLANNING

How to succeed in business without really frying

By now you will have read the section on 'Attacking the problem'. You should have certified your software and data and backed up everything that moves. Now you can sit back and watch the city burn around you, right? You've forgotten Murphy's Law: 'if anything can go wrong, it will'. You still need to be prepared. In order to do this satisfactorily, you must identify key components of your business and protect them.

What are key components? They are parts of your business without which you cannot do business (or at least cannot do business efficiently). A salesperson's price list and a tradesperson's mobile phone might qualify for this category. You need to look carefully at the way you do business and identify areas which could cripple you. Having identified such areas you need to ask 'can I protect this area?' or 'can I work around a failure in this area?'.

Consider the following questions:

- Do you rely heavily on a PC for transactions or data? It might be a good idea to protect the PC from power failure using an Uninterruptible Power Supply (UPS). Depending on the quality of the unit, a UPS will run for anything from a few minutes to several hours during a power cut. If your business cannot function without a PC then you should probably have a UPS — not just for Y2K but for any power fault. However, a UPS won't protect you from program bugs that may become evident after a Y2K date horizon.

  Another approach is to plan alternative ways of working that don't involve the PC. Print out hard copies of things such as price and phone lists. It may be a slow process looking up information manually, but it's better than having no access at all. If you take bookings via a computer, a printout of current bookings may help you to keep working until service is restored.

- Do you rely on regular supplies for manufacturing or some other activity? What assurances do you have from your suppliers about their ability to continue to supply you? If possible, try to get assurances from all critical suppliers. Consider whether there are alternative suppliers or items which can be substituted if your current supplier cannot deliver on time. Consider building up a stockpile of essential supplies.

- Have your premises been checked? Do your business premises have a security system? Do they have electronic locks, lift controls or alarm systems? When you first come into work, next year, will you be able get in? Will your customers have access?

An article in *Business Review Weekly* ('Year 2000 still holds a few shocks' by Ross Langeford, March 1, 1999) suggests that many building owners are completely ignoring Y2K issues, while others are dealing with it in incomplete ways. In one case described in the article a building had been inspected and the lift system certified compliant, but the alarm system was not compliant. Disabling the alarm system disabled the lifts.

A high proportion of building owners who have conducted testing have located significant problems. It is reasonable to expect many buildings will have access or security problems, and it's a good bet that service and repair staff may be over-booked for January 2000.

It would be wise to ask some hard questions of your building owner — doubly so if the owner is yourself. If disruption appears a possibility, it may be wise to investigate alternative premises. In some industries it may be possible for staff to work from home.

Each business will have its own unique areas of vulnerability and you must look carefully at your own to identify such areas.

MAKE SURE YOU KEEP RECENT BANK STATEMENTS AND RECEIPTS FROM YOUR MOST RECENT TRANSACTIONS

FINANCIAL CONTINGENCY PLANNING

Assume the crash position?

Please do not regard this section as a substitute for professional investment advice. You should review any advice critically and, as far as possible, get unbiased, independent information. Understand that asking your bank manager whether to leave your money in the bank or your stockbroker whether to sell all your shares may not constitute unbiased advice! Beware.

Some advisers are suggesting that you withdraw all of your money from banks. This isn't necessarily a good idea. Obviously, if large numbers of people start hoarding cash, there would probably be a great increase in the incidence of theft and people with cash would be vulernable.

It's highly unlikely that any of our major banks will stop operating, and even if they did they are guaranteed by the Reserve Bank. If the Reserve Bank ceased operations, then cash would probably be worthless anyway. If you assume that banks will not go under, the major risks are temporary loss of service and incorrect transaction records. It's reasonable to have *some* extra cash on hand as a precaution against ATM malfunctions on January 1.

EFTPOS will depend on the reliability of bank computers and telecommunications; in Australia there is a fairly low risk of failure. Ordinary credit cards will probably work fine, even if the EFTPOS systems are down, though there may be limits on the size of transactions if vendors cannot get approvals. Cheques will, of course, work as normal — but not everybody accepts cheques.

Make sure you keep recent bank statements and receipts from your most recent transactions, just in case there is a dispute about your balance.

Some commentators consider there is a very high risk of a stock market crash in December 1999 as nervous traders sell in anticipation of Y2K. This creates a secondary risk that the market could crash before December 1999, due to traders attempting to get out. There is also a fairly high probability that the market will rebound in January as traders jump back in. There will be much money to be lost, and possibly much to be made. It's certainly a time to make sure you get good advice, rather than speculating.

PERSONAL CONTINGENCY PLANNING

How to learn to stop worrying and love the bomb

Personal contingency planning is primarily planning for your safety and comfort — we're talking here about preparation for all possible emergencies, not just Y2K. For most people, this is relatively simple:

- Stockpile a few days worth of water and non-perishable food, just in case. Even if there is no loss of supply (and the chances are there won't be) there could be artificial shortages created by too many people trying to stock up.

- Make sure you have enough cash for your taxi fare home.

- Get your investments in order.

- If you want to program your video recorder to tape something after midnight, then set the date back to 1998 just to be sure.

Beyond this it isn't too complicated, unless you live on the 49th floor of a building with a non-compliant lift (if so, you'd better start those extra aerobics classes pretty soon!).

There are plenty of doomsayers who will seriously recommend holing up in a bunker with a year's supply of food and some heavy-duty firearms. In Australia, this would be an overreaction. While the potential exists for some serious interruptions to our lifestyle, the likelihood of TEOTWAWKI (the end of the world as we know it) is just about nil.

However, Y2K could be much more hazardous for people with special needs. For a person with reduced mobility, such as a quadriplegic or an elderly person, a lift breakdown could be a serious problem. Loss of computer or phone services could mean a blind person is cut off from support. If you're on any medication, interruption to its supply could put you at serious risk.

If you have reduced mobility or are visually impaired, make sure you have a contingency plan on how to get into and out of your home without power or other services.

If you rely on medication, make sure you've got a good stockpile.

- If you rely on medical equipment, whether it's a pacemaker or a dialysis machine, make sure it's certified compliant. If possible, make a fall-back plan, such as use of a dialysis machine at a local hospital.

If you need a service to maintain your health or safety, then even the remotest chance of failure must be taken seriously.

GOVERNMENT CONTINGENCY PLANNING

Y2K? What Y2K?

Perhaps the most serious area of consideration is the potential failure of government systems. While all Australian government departments are making concerted efforts to prepare for 2000, public information about their progress is a little sparse. Some are way ahead, a few have started rather late and others may be hampered by the sheer complexity of their systems. Some observers expect large-scale disruption to many government services. If the performance of tax audits, or the issuing of speeding tickets is disrupted, most of us will be able to contain our disappointment, but if welfare payments, health or transport services suffer, the consequences may be more serious.

EVIDENCE SUGGESTS THAT AN ALARMING NUMBER OF JAPANESE BUSINESSES ARE SIMPLY IGNORING THE PROBLEM

If you receive a pension, unemployment benefits or any other welfare benefit, don't rely on being paid on time in January. That doesn't mean you won't be, but Y2K brings an added risk. It might sound unreasonable to tell pensioners and the unemployed to start saving up, but our advice about having a little extra cash on hand applies here, too.

And if your business relies heavily on government contracts, you might want to be very careful about your cash flow around Y2K time. Getting paid for government contracts has never been easy, but if their computer systems fail it will be even harder.

INTERNATIONAL CONTINGENCY PLANNING

It's a small world, after all . . .

Many analysts agree that Australia is one of the best prepared countries in the world. This doesn't imply for a second that we won't suffer any problems — we will. However, the problems are likely to be temporary and we will almost certainly deal with them. Some businesses may go to the wall, but most will survive and prosper.

Around the world the story may be different. The US seems relatively well prepared, though it is also expecting some serious disruptions. European Union countries have been heavily concerned with their currency changeover. Many, such as Italy and Germany, appear to have devoted inadequate effort to Y2K issues.

Due to recent economic turmoil in the region, Asia is said to have Y2K way down on its priority list. Japan, in particular, is a source of great concern. Evidence suggests that an alarming number of Japanese businesses are simply ignoring the problem, and analysts are predicting major disruptions and possible economic collapse. If the Japanese economy suffers major damage, it seems inescapable that the rest of the world won't feel the aftershocks, and Australia will be quite exposed.

Information from South-East Asia has been a little sketchy, but some analysts believe that major failures are imminent. In China, where it is estimated that 90% of all software is pirated, firms are reportedly hav-

If Excel crashes in January, you can be fairly confident that Microsoft will have a patch or an upgrade within a fairly short period. If it's your own baby, then nobody is going to fix it but you. You might want to make your programmers wear pagers, or better yet hold a *compulsory* New Year's Eve party at the office (though you'd better make sure they don't spike the punch). However, you'll probably find that a gram of planning goes further than a kilo of panic.

ing difficulty with Y2K remediation because they cannot get vendor support for pirated software. Failures in China and many other Asian countries would be less disastrous for the general population than in a country like Australia, as they are less dependent on technology; the impact would be greater in cities than in rural areas. However, the main victims might well be the burgeoning high-tech industries which are central to so many Asian economies. Damage to these industries, many of which are still recovering from economic problems in the region, could also have serious effects on Australia.

SOFTWARE CONTINGENCY PLANNING

To err is human, but to really mess things up requires a computer.

If you have software that was developed inhouse, you face some additional challenges.

- Make sure everything is documented. When an application fails you need to know which application it is, where to find the source code and which programmers are familiar with it.

- Make sure that your software tools are Y2K compliant. January 2000 is not the time to discover that your debugger has Y2K problems. Contact tool vendors.

- If you are a programmer, make sure that you can rebuild your software. It's alarmingly common to find that, when you try to rebuild an old application, it will no longer compile. The new version of the compiler may be incompatible, a library or other resource may have changed. Again, January 2000 is not the time to discover this.

- Watch out for side effects. A good definition of a software upgrade is 'the act of trading your old bugs for a new set'. When you make a quick Y2K patch to get around a problem, watch out that you don't introduce an even bigger problem. No matter how much of a hurry you're in, some careful analysis can often avoid side effects resulting from changes you thought insignificant. Make *absolutely* sure you test the software properly before you start to use it. Haste makes waste!

TRIAGE CONTINGENCY PLANNING

When the sky is falling

The term 'triage' was first used in a Y2K context by Canadian consultant Peter de Jager (see page 26). In his paper, *Systemic Triage*, de Jager describes the original definition of the word thus: "The sorting and allocation of treatment to patients, especially battle and disaster victims, according to a system of priorities designed to maximise the number of survivors".

At a certain point, some businesses are going to reach the realisation that they're not going to make it. No matter how many dollars and working hours they throw at Y2K remediation, they're going to miss their deadlines.

At this stage you should stop trying to solve the problem, and concentrate on harm minimisation. This is a painful decision, and one that will be resisted. However, once it is plain that you're not going to make it, the quicker you refocus on triage issues, the better the chances that the business will survive — 'survive' being the operative word. If you cannot do business during January 2000, then you are probably on the road to a quick and gruesome bankruptcy.

The first step in triage planning is to take that hard look at your business which we have already described. What are its core functions? What tasks are so essential that without them the business cannot operate? It is these tasks, and only these, on which you must concentrate the triage effort. For each system you must answer the questions:

- Can this system be patched in such a way that it continues functioning? This may mean some features no longer work. For instance, automatic scheduling of operations may no longer work; you may have to manually operate equipment that was previously handled automatically.

- Can you work around the system's flaws using manual methods? This may mean invoices have to be amended by hand in order to keep shipping products, or shipping addresses have to be typed up in a word processor.

- Could the system be replaced with an off-the-shelf package? These are widely available for accounting systems, inventory systems and other common tasks. A system that does most of what you need, correctly, may be preferable to one that does everything you need, incorrectly. Identify such packages now and make sure you have enough time to implement and customise them.

ACT NOW

The time has come, the walrus said . . .

If you haven't yet completed your contingency planning then you should get to it *now*! Good contingency planning takes time, and implementing that plan will take more time. If you plan to stockpile food, components or anything else don't start in December — everyone else will be doing it then and you might just miss out. Don't panic, be sensible.

Checkpoint — Plan of action

- Everyone will need to take *some* precautions
- Back up as much as possible, but at the very least your hard drive or data files, and remember to include recent work
- If you choose floppy disks to back up, make two or three copies of each
- Identify the parts of your business without which you cannot continue operating
- Once identified, work out how best to protect those areas, or plan workarounds
- Consider using a UPS with your system
- Test all software thoroughly and document all the programs you use
- Make sure you can contact the programmers if any software was written inhouse
- Print out hard copies of information with which you can keep your business turning over if your PC does fail
- Stockpile essential business supplies if you are not certain your suppliers will be able to deliver
- Check your premises will be accessible and secure
- If you realise you won't be Y2K ready, at the very least, ensure the core functions of your business can still operate.
- Have *some* extra cash on hand as a precaution on January 1, to cover both ATM failure and lack of pensions or benefits
- Keep recent bank statements, and receipts from most recent transactions
- Stockpile a few days worth of water and non perishable food
- People with special physical needs should stockpile medication and make arrangements for assistance if it should be required
- Make sure you organise any extra supplies, either of cash or goods, well before the end of the year

Public relations — letting it all hang out

THERE IS A GREAT RELUCTANCE among many companies to part with any information concerning Y2K readiness. In this climate of fear, they are worried that information which is unfavourable, or even favourable information that might be misinterpreted, could lead to panic. But the resultant lack of information feeds the fears and the climate gets worse. If we make information available, warts and all, then we are better able to realistically assess the risks and address people's fears. Share prices have actually been known to *rise* when a company admits it has Y2K problems, because it gives the shareholders confidence that the company must be addressing the issue.

The Australian Stock Exchange (ASX) has required Y2K status reports from all listed companies. At first, many of these reports were so vague as to be of little use, but they have improved in recent months. Many companies are reluctant to claim compliance, because it might open them up to legal action if they've overlooked anything. This makes it hard to distinguish well prepared, but reticent, companies from those which are being deliberately vague to hide their impending disaster.

THE JERICHO ROAD

In February 1999, the Federal Government enacted the *Year 2000 Information Disclosure Bill 1999*. Modelled on the so-called 'good Samaritan' legislation from the US, this law is intended to encourage disclosure of Y2K information by reducing the fear of getting sued. According to a statement from Senators Alston and Campbell, the bill can:

- protect a person making a Y2K disclosure statement from civil liability arising from the making of the statement — for example, for negligent misstatement, defamation or liability under trade practices and fair trading legislation — subject to certain exemptions;

- provide that a Y2K disclosure statement will not be admissible as evidence against a person who made it — for example, the statement will not be able to be used in legal action resulting from the failure of Y2K goods and services;

- provide that the exchange of Y2K information will not give rise to liability under section 45 of the Trade Practices Act, which prohibits certain anticompetitive contracts, arrangements or understandings;

- offer these protections from the enactment of the legislation until June 30, 2001.

To qualify for this protection, the statement must:

- be clearly identified;
- be in writing;
- relate solely to Y2K processing issues;
- identify the authoriser.

The legislation will not provide protection if the statement:

- is known by the maker to be materially false or misleading;
- is made recklessly;
- is made in connection with the formation of a contract;
- is made in fulfilment of an obligation under a contract or law;
- is made for the purpose of inducing customers to acquire goods;
- relates to restraining injunctions or applications for declaratory relief;
- relates to civil actions undertaken by regulatory bodies such as the ACCC [Australian Competition and Consumer Commission]; or
- relates to civil actions relating to the infringement of intellectual property.

Critics of the bill have said that it provides too little protection, and that it has arrived too late to do any good. It may also take some time for companies to understand the bill and take advantage of the protection it offers. A number of companies were still lax in reporting to the stock exchange by its March 31, 1999 Y2K disclosure deadline.

INTERACTIONS

Let's presume that a company is, internally, 100% prepared for Y2K. If a supplier for a critical component of that company's product collapses, then the company's collapse may soon follow. Alternatively, if a big customer for that product collapses and cannot buy any more (or worse still, cannot pay for product already delivered) then, again, the company's demise is quite likely.

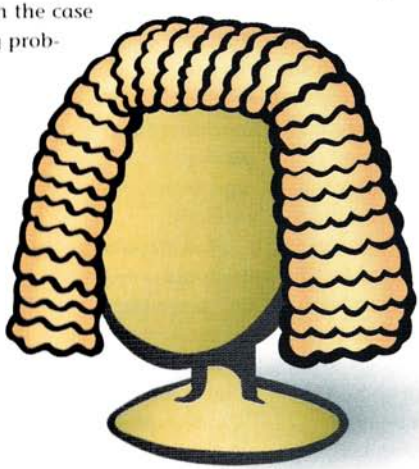
Such relationships may make or break many companies next January if Y2K has a serious impact. We need to take steps to ensure that we're not victims, just as we need to ensure we're not culprits.

The main responsibility in not becoming a culprit is to stay in business. Remain solvent, keep buying and selling (and of course paying bills). If there's a problem, be honest about it and give suppliers and customers time to make arrangements: "Do you mind if I ship next month's product early, because I'm expecting some problems in my dispatch system?"; or perhaps "I'm expecting a slump in the first quarter of 2000, so only manufacture half as many widgets for me".

Not becoming a victim is more difficult. Where possible, obtain information about suppliers and customers. This is especially true for customers who might owe you money. Investigate alternative suppliers, in case the worst happens. If possible, consider stocking up on any critical components.

Insurance and litigation

IT'S A COMMONLY HELD BELIEF that no insurance company will ever cover Y2K disasters. While it's true that many new policies have clauses limiting the insurer's liability in the case of millennium bug problems, recent statements from the Insurance Council of Australia have indicated this isn't a blanket refusal. If you're willing to pay the premium demanded and demonstrate that you have done everything possible to make your systems compliant, many insurance companies will be happy to help you . . . and will make sure you continue to pay through the nose well beyond New Year's Day.



Note that if an insurance company makes any changes to the conditions of an existing policy, it must, by law, notify clients of the new arrangements, so Y2K cannot be excluded without your knowledge. Existing business continuity insurance is unlikely to cover millennium issues.

The possibilities of Y2K litigation have generally taken a back seat in the race to prepare computers and businesses for the end of the year. It's been said that lawyers are rubbing their hands in glee at the thought of so much potential business, and in the US it has been predicted that Y2K litigation might dwarf the consequences of the bug itself. But current Y2K lawsuits — where an organisation has invested heavily in new equipment and then found it is not millennium-compliant — are generally still test cases.

After this year, it is expected that cases will emerge where the litigant is a person or organisation which has suffered a loss due to someone else's Y2K misfortunes. Loss adjustment firms have predicted that small to medium enterprises may be in the most difficulty after January 1, having failed to prepare sufficiently and therefore caused losses for those which rely on them. Ignorance will be no defence.

For example, if Y2K problems meant you were responsible for a delay that caused another party to miss a flight and an opportunity to close a business deal, you could be vulnerable to legal action.

The advice here is no substitute for that of a Y2K lawyer, but to defend yourself in such a situation you would need to show

that you had exercised the degree of care and diligence that any reasonable person in a similar position would have exercised. You need to take adequate steps to try and eliminate the Y2K problem and, should it become apparent that you cannot fix everything, to take adequate steps to limit its impact.

Corporations Law requires that company directors disclose in annual reports any matters that have arisen since the last financial year's report that may significantly affect the company's operations. This, along with the 'good Samaritan' legislation passed by the Senate, and the advice offered by the Australian Stock Exchange to its member companies, means there is a healthy environment for full disclosure by all Australian companies.

The complex interrelations behind computer failures mean that from the victim's perspective it can be difficult to work out which party to sue, and whose insurance should cover the damage. It isn't not worth jumping up and down unless you're absolutely sure what happened. A system that fails may induce a secondary event more damaging than the original failure — such as a fire caused by a climate control system that is malfunctioning due to Y2K problems in other building systems — but the cause may initially be hidden by the results. A company may also have provided a product containing components produced by a number of suppliers, none of which is compliant. From the buyer to the original component supplier, each will blame the other.

It will be difficult for software or hardware providers to claim that Y2K is an 'act of God' or 'force majeure' because they created the item that is producing the fault. But it may be possible for them to prove that data-related problems are the fault of the user, as long as they are not directly connected with a function of the program or machine. This will not prevent suppliers placing 'act of God' clauses in their licence agreements, so make sure you read them carefully.

For the same reasons, it's hard to say that once a software or hardware provider sells a product, it continues to have a duty of care — it depends on how the product is used. Hamish Fraser, of McCullough Robertson Lawyers, has written perhaps the definitive Australian paper on the subject (see http://www.iib.qld.gov.au/Y2K).

He points out that it may not be a valid move for a software company to issue a warning in 1998 about the impending failure of a product it sold in 1994 because the customer should have been made aware of the flaw in the first place. However, the law may regard the program as having been sold in good faith at a time when Y2K issues were not yet understood.

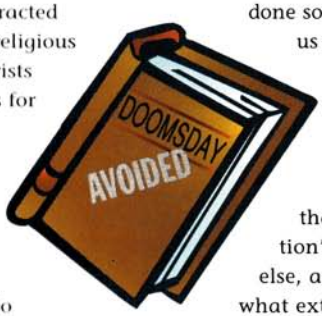
Test cases will resolve whether the computer industry as a whole has been ignorant and irresponsible, or whether the complexity of Y2K is actually something we couldn't have foreseen. Business is always risky and financial year 1999/2000 will be especially so. However, care and planning can go a long way.

Conclusion

THE WORLD'S ORIGINAL Y2K GURU, Peter de Jager, came to prominence with an article called *Doomsday 2000*. Recently he's taken to saying that we have broken the back of the Y2K problem and that while there is still much to be done, the potential for doomsday has been avoided.

For this statement he's taken a lot of flak — plenty of people are engaged in making money out of others' Y2K misfortunes (real or imagined) and they don't like de Jager calling them opportunists or damaging their cause. Y2K has attracted more than its fair share of religious extremists, conspiracy theorists and bogus technical experts for whom a computer-related crisis combined with a very significant date is the perfect mechanism to make people employ or follow them. Increasingly, the media is falling victim to such hype merchants and it seems possible that unfounded panic in the streets might become a threat in addition to the technical Y2K problems. It is everyone's responsibility, therefore, to spread reliable information, countering the Armageddon myths.

EVEN IF YOU ARE READING THIS BOOK AFTER 2000 IN THE MIDST OF AN EMERGING CRISIS, YOU CAN STILL DO SOMETHING



But as de Jager pointed out, the situation is very far from all clear. There is still a lot of work to be done, and you probably have a lot to do. Y2K could still be a major disaster for many people and even if you aren't one of them, their problems may cause ripple effects that should concern you.

If you haven't yet addressed Y2K, don't be afraid to begin now. Yes, you should have done something long ago, but it isn't a missed opportunity. Even if you are reading this book after 2000 in the midst of an emerging crisis, you can still do something. Plan around it, fix it, or scrap it, and you will have done something to reduce the risks for us all. Get to work now, with our Pocketbook CD as your initial toolkit, and refer back to the information on these pages as required.

The real Y2K problem is that each person or organisation's readiness affects everyone else, and nobody is exactly sure to what extent. Most large organisations have the resources to fix things, so most now present minimal Y2K risks, but smaller businesses and individual citizens have been slow to act. Their combined troubles could be a big problem for society as a whole.

Another possibility is that if enough work has been done, nothing significant will happen when 99 turns into 00. No doubt

the media will scream that it was all a furphy and the credibility of people like de Jager will be challenged. But the truth is that if that does happen, we will owe a great deal of thanks to de Jager and many others (even some of those annoying, ever-so-wealthy consultants).

It's possible that we'll never know the real scale of what might have occurred if nobody had carried out any remediation. Wise people will be able to say that Y2K was real, while also denouncing those who dealt in exaggerations. Hype merchants are on thin ice, because the general population is quickly becoming as literate in date-related computerspeak as it is about the Internet. It takes time, but eventually people understand the possibilities and limitations and incorporate it into their world view.

THE REAL Y2K PROBLEM IS THAT EACH PERSON OR ORGANISATION'S READINESS AFFECTS EVERYONE ELSE

The New Year's Eve celebrations for 1999 will be tinged with nervousness. Parties on the streets will contrast with the hard work going on in IT departments in the buildings above or among the ranks of the emergency services (who are professional contingency planners), and perhaps with groups expecting alien landings or apocalyptic events.

Some people will have absolutely no time to ponder their New Year's resolutions, which is a pity because technology industries owe the world a greater degree of responsibility than they have shown by creating Y2K in the first place. On the other hand, pointing the finger is a waste of valuable remediation time — the responsibility for getting out of the year 2000 hole rests on all our shoulders. In the unlikely event that things do get out of hand on New Year's Day, the best thing you can do is sit tight, enact your contingency plans and wait for advice from the authorities. Y2K affects the government just as it affects the people, so you can be sure that emergency plans will fall into place if needed.

There will no doubt be some potential catastrophes that follow Y2K and are compared to it (just as every cyclone or earthquake is compared to the last), and a pattern will have been set for publicising and dealing with such crises. Whether they're as serious as this one or not, the best attitude is to consider the consequences and be ready for what may come. Businesses in particular will learn new continuity skills — hopefully this will make the world a more stable place.

Stay calm, address the Y2K issues that affect you, and you will have helped not only yourself, but those who depend on you. The earlier you do this, the better, but it's never too late to start.

Good luck, and here's to a happy new year!

y2k

Information

Internet resources

NEWS

Year2000.com
Peter de Jager's world-leading, level-headed news and information site. The best place to start learning about Y2K, and always worth another visit.
http://www.year2000.com

Y2K News Magazine
A spin-off from Year2000.com, looking at Y2K-related headlines in the global online media. Regularly includes Australian coverage.
http://www.y2knews.com

Y2K Today
News and background information on everything to do with Y2K. A good counterpart to the de Jager sites.
http://www.y2ktoday.com

ZDY2K
Ziff-Davis' dedicated Y2K news and information site. A thorough, down-to-earth and usually reliable source of information.
http://www.zdy2k.com

Duh-2000
A monthly contest for the stupidest thing said about Y2K. Beware, they may be watching you!
http://www.duh-2000.com

STATE GOVERNMENT Y2K SITES

New South Wales Year 2000 Home Page
Originally the combined Australian governments' Y2K resource. A good place to start, whichever state you live in.
http://www.y2k.gov.au

Victorian Government Millennium Bug Awareness
A bright, chirpy site with practical information for Victorian businesses.
http://y2k.dsd.vic.gov.au/

Queensland Government and Y2K
A comprehensive information resource for Queensland residents.
http://www.y2k.qld.gov.au

Y2K South Australia
Under the watchful eye of SA Minister for Year 2000 Compliance, Wayne Matthew.
http://www.y2k.sa.gov.au/

Western Australia Year 2000
Includes an overview of what action the WA government is taking.
http://www.y2k.wa.gov.au/

Northern Territory Year 2000
Complete with information on seminars being held throughout the territory.
http://www.nt.gov.au/year2000/

INFORMATION

year2k Industry Program
The home page of the Federal Government's year2k Industry Program, targeting small to medium businesses (SMEs).
http://www.year2k.com.au

Y2K Register
This site provides information on products, services and Y2K compliance in Australia.
It's run by the year2k Industry Program and Standards Australia.
http://www.y2kregister.com.au

Institute of Engineers Y2K directory
A professional information

site which is dedicated to helping you track down engineers with Y2K knowledge.
http://www.ieaust.org.au/y2k/

Y2K Compliance Database
A searchable index of the Y2K readiness of software, hardware and consumer goods.
http://www.Y2Kbase.com

AIIA Year 2000 Computer Problem Information
The Australian Information Industry Association's take on the bug.
http://www.aiia.com.au/Year2000information.html

OGIT Year 2000 Project Office
Check how the Australian government is preparing for 2000 via the Office for Government.
http://www.ogit.gov.au/year2000/

Year 2000 Risk Assessment and Planning for Individuals
Gartner Group's research paper advising how each individual should prepare for January 1, 2000, and beyond.
http://gartner5.gartnerweb.com/public/static/home/00073955.html

The Millennium Problem in Embedded Systems
Professional advice and research on embedded chips provided by the British Institution of Electrical Engineers.
http://www.iee.org.uk/2000risk/

Taskforce 2000 — Contingency Planning
A comprehensive British article on Y2K contingency planning for individuals and business.
http://www.taskforce2000.co.uk/contingency.htm

The Crouch-Echlin Effect
Mike Echlin's rough and ready research on the controversial Crouch-Echlin time dilation effect alleged to affect PCs after 2000.
http://www.intranet.ca/~mike.echlin/bestif/index.htm

PEOPLE

Karl Feilder
The man who invented the "five level" theory of Y2K.
http://www.feilder.com

Ed Yardeni
Deutsche Bank's chief economist and Y2K recession forecaster.
http://www.yardeni.com

Ed Yourdon
Mainframe programmer and Y2K doomsday theorist.
http://www.yourdon.com

Dr Leon Kappelman
The Y2K expert who explains there will never, ever be a silver bullet.
http://www.year2000.unt.edu/kappelma/

PRODUCTS

Viasoft — OnMark 2000
http://onmark.viasoft.com

Symantec — Norton 2000
http://www.symantec.com.au

Greenwich Mean Time — Check 2000
http://www.gmt-2000.com.au

Prove It 2000
http://www.proveit.com.au

MFX Research — MFX 2000
http://www.mfxr.com

Glossary

BIOS: Basic Input/Output System. The initial settings that tell a computer what to do when it starts up, stored in read-only memory (ROM).

CMOS: complementary metal-oxide semiconductor. Where variables used by the BIOS are stored.

Cobol: Common Business Oriented Language. An aging programming language used with mainframes.

Contingency: An event that may or may not take place. Contingency planning involves being ready for what might happen.

Domino effect: Where one event causes another; a chain reaction like rolling dominoes. In Y2K terms, where a Y2K problem triggers more serious events in related processes.

Embedded system: A very basic computer, often just a single chip with limited instructions, used where a full computer would be overkill.

Good Samaritan: From the biblical parable where a Samaritan helped a wounded man. In Y2K terms, the Australian government's legislation to help companies declare their Y2K compliance.

Mainframes: Large-scale centralised computers connected to dumb terminals, typically in big organisations.

Murphy's Law: If anything can go wrong, it will.

NOS: network operating system. The software environment which helps run a network.

OCR: optical character recognition. A technique for identifying text on a scanned page.

OS: operating system. A software environment which helps run a computer.

Pivot date (or windowing): Where years denoted in two digits are presumed to fall within a specified 100-year period (such as between 1930 and 2029).

Remediation: The Y2K industry's favourite word for the Y2K repair process.

RTC: Real-time clock. The battery-operated clock in a computer, used by the BIOS to figure out the time and date.

Silver bullet: A silver bullet is said to be one way to kill a vampire. In Y2K terms, a mythical program that will automatically fix Y2K in one swoop.

TEOTWAWKI: The end of the world as we know it. In Y2K terms, the theory that our reliance on failing computers will cause civilisation to collapse.

Triage: The sorting and allocation of treatment to patients, especially battle and disaster victims, according to a system of priorities designed to maximise the number of survivors. In Y2K terms, this means doing the same with affected systems.

Workaround: An alternative solution aimed at avoiding any process susceptible to problems.

The Pocketbook CD and disclaimer

ALL THE TOOLS AND DOCUMENTS you need to help you get ready for Y2K.

FREE BIOS TESTING AND FIXING PROGRAM

Featured on the CD is a free, full working version of Viasoft's BIOS Test & Fix software, which will allow you to test your BIOS and, if necessary, install a fix to make it Y2K compliant.

BIOS TESTING TOOLS

Test your BIOS clock's ability to cope with the millennium date rollover and related problems. A selection of applications that give detailed reports and recommendations about your hardware's Y2K compliance and what you need to do about it.

Y2K REPAIR TOOL DEMOS

A collection of tools with the capacity to diagnose and repair your PC's Y2K problems. These demonstration versions subject your system to a detailed examination beyond the BIOS, helping to sort out your operating system and applications. Many integrate reports and recommendations from software companies,

PATCHES

Y2K service packs and patches for operating systems and many popular applications.

COMPLIANCE INFORMATION

Information and links to more on the Web. Extensive Y2K compliance information tables are included, as first published in *Australian PC User*, February 1999. The tables contain advice from software companies regarding the Y2K readiness of their products, covering commercial packages, shareware, operating systems and even non-PC environments. Plus a selection of the best white papers and essays from experts in the Y2K field.

DISCLAIMER

The Y2K Pocketbook and CD are a guide to help you identify and fix Y2K problems in the home in a small business, but it is by no means a complete guide or a final solution to the problem. We make no express or implied warranties that this Pocketbook will identify or fix your Y2K problems. Use the Pocketbook and CD as a starting point in getting your PCs ready for 2000.

The advice in this Pocketbook is accurate to the best of our knowledge. However, *APC* cannot and does not accept liability for any loss (including any financial loss) arising as a result of following the instructions in this book. In the event that such a liability cannot be excluded by law, liability is limited to replacing this Pocketbook or refunding its purchase price at our option.

In order to save you time and money downloading the Pocketbook CD contains programs freely available on the Web. The software on the cover CD is provided as is, with no expressed or implied warranties, and *APC* cannot be held liable

for any damage that may incur as a result of using the software or documentation provided. As with all software downloaded from the Web you should read installation instructions and licensing agreements carefully and scan for viruses all programs downloaded from the CD. While the **apcmag.cd** team do their best to ensure all software on the CD is checked for viruses, new viruses are discovered all the time and as a result you should scan all software using a virus scanner with the latest antivirus updates. You should also back up any important data, as we can make no guarantees that software, downloaded from the Web as it is, will operate in accordance with the program author's intentions.

As the software is freely given and not authored by *APC*, neither *APC* nor the author of a given piece of software (unless explicitly stated) can provide technical support for the programs contained on the Pocketbook CD.

If the Pocketbook CD is faulty simply contact us on (02) 9288 9123 and we'll send out a replacement CD.

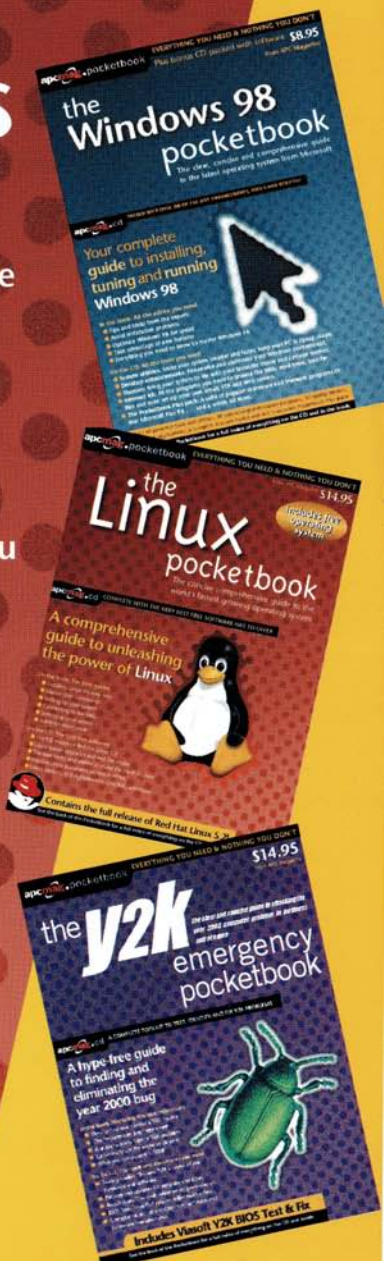
EVERYTHING YOU NEED AND NOTHING YOU DON'T

apcmag.pocketbook

Pocketbooks

Pocketbooks offer you all the facts without the padding, at an affordable price. Each Pocketbook is a valuable, concise and entertaining resource, and comes with a cover CD packed with all the best software.

The Pocketbook series: Everything you need, and nothing you don't.



The Windows 98, Linux and Y2K Emergency pocketbooks are on sale now at newsagencies, and can be purchased online at apcmag.com/shop or phone (02) 9260 0000 or toll free on 1800 252 515.

Look out for The Networking Pocketbook, available soon. Future titles include the Revised Edition Linux, Uprgrading, Windows 2000, Webmasters, Digital Imaging pocketbooks, and many more . . .

If you're a home user or run a small business you still have time to tackle Y2K before it tackles you. But you have to act now. Right now. The Y2K Emergency Pocketbook will help you understand the problem and show you how best to deal with it.

Inside this Pocketbook:

- **Y2K: A history**
  Learn how the Y2K problem came about, why it is posing a threat to so many people, and why there is no quick-fix solution.
- **The bug in its many forms**
  Read what the experts have to say about the phenomenon. It isn't just your computer system that poses a threat. Also at risk are embedded systems. What are they and where do they occur? What about other people's Y2K problems? Can they impact on your business? How might this happen and what should you be aware of?
- **Attack, attack, attack**
  Act now, or forever hold thy peace. What should you be looking for? Learn how to pinpoint the areas that are likely to cause you problems and what repair process you should undertake.
- **Contingency planning**
  How do you prepare for problems if you don't know what they'll be? Find out how to get to grips with what may or may not happen. And what are the legal implications of not planning well enough ahead?
- **Checkpoints**
  Where appropriate, checkpoints have been provided to offer you an overview and to allow you to check at a glance what steps you should be taking.
- **Not only, but also**
  A comprehensive list of Web sites that will enhance your understanding of Y2K.

On the Pocketbook cover CD:

- **Viasoft BIOS Test & Fix**
  Free! Full working version that actually makes your BIOS Y2K compliant.
- **Professional millennium bug toolkit** Track down problems with the latest Y2K repair tool demos. Choose the best bug fix for your needs.
- **Operating system upgrades**
  Stop Windows crashing in 2000. Install the latest patches for your OS and reach the recommended level of Y2K safety.
- **Software patches**
  Save your software. Official Y2K service packs included for many popular applications.
- **Happy new millennium!**
  Fun with Y2K: screensavers, countdown clocks, desktop themes and bug games galore.
- **Information**
  Y2K compliance advice from hardware, software and shareware companies. FAQs and links to more.
- **Plus: Microsoft IE 5.0, Netscape Communicator 4.51, Netscape Communicator 4.08, Netscape Navigator 4.08 and all the add-ons you need to navigate the CD.**

ISBN 1 876587 06 7

9 313006 006039 05